2019

EXPERIMENTAL CHARACTERIZATION AND SCALE MODELING OF A VERTICAL AXIS WIND TURBINE

Milo Ferrazzoli
University of Rhode Island, otis728@icloud.com

EXPERIMENTAL CHARACTERIZATION AND SCALE MODELING OF A VERTICAL AXIS WIND TURBINE

BY

MILO FERRAZZOLI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING AND APPLIED MECHANICS

UNIVERSITY OF RHODE ISLAND

2019

MASTER OF SCIENCE THESIS

OF

MILO FERRAZZOLI

APPROVED:

Thesis Committee:

Major Professor David G. Taggart

Carl-Ernst Rousseau

Christopher Hunter

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2019

ABSTRACT

A novel method of harnessing wind energy utilizes fixed surrounding stators which converge on a vertical axis turbine. Destructive fluid interactions cause inconsistent starting and diminished efficiency of such a design. In this research, experimental performance analysis of scaled turbine geometries is used to evaluate design parameters and predict turbine behavior, in order to minimize these effects.

Purely vertical turbine blades paired with parallel stators create a passing interference along with points of force balance during low wind conditions. The varying low-wind torque outputs of such states decreases the predictable starting of the proposed design. This adds further complication to anticipating the turbine size needed to start a preselected generator with known breaking torque.

One method of reducing symmetrical loading is to add a helical form to the turbine. Another is to increase the distance between stator and turbine tips within the windward cavities. The latter being achievable by allowing a portion of the stator to pivot away from the turbine in the direction of rotation.

To analyze the performance of varied geometries, scale turbine replicas, with a height of 2.5 and 5-inches, were constructed. Each was designed to fit within the Aerolab wind tunnel of the University of Rhode Island Mechanical Engineering Laboratory. The 5-inch model was built with interchangeable stators and turbines allowing for various combinations of open or closed stators with a range of turbine helices. The output shafts were equipped with a load cell or optical encoder paired with custom data acquisition hardware and software.

The static performance of different stator and turbine combinations was measured by fixed-turbine torque readings at various starting angles. Rotational performance was quantified by measuring angular change of the turbine released from rest and free moving in constant and ramped wind conditions. Dimensional anal-

ysis was applied to the gathered data in order to develop torque scaling equations.

The replica testing indicated that the combination of open stators with a 45° top-to-bottom offset helix provided the least propensity for stalling, along with the greatest acceleration, top speed and static torque generation. These results were scaled to predict that a 5 kW generator could be started by a 7-foot-tall turbine exposed to a 2 m/s wind velocity. The prediction was then validated by testing a full 7-foot prototype.

ACKNOWLEDGMENTS

I would like to extend my deepest gratitude to my major Professor and advisor, Dr. David G. Taggart. From class to research projects, Dr. Taggart has always expressed great patience and care in the higher education of all his students and this experience followed suit. Dr. Taggart has been an amazing mentor and wealth of knowledge through the proposed research and all projects I have been graced to be a part of. I would like to say a great thank you for the lengthy support you have provided and for bearing with me through the extended time to complete this research.

I would like to thank Dr. Carl-Ernst Rousseau for contributing his time and expertise towards the understanding and completion of this project. Dr. Rousseau was an outstanding resource for ideas and reassurance of strategies. Thank you for the continued support as a long-standing inside committee member.

I would like to recognize the incredible resource that Dr. Stephen A. Jordan was during the early stages of this project. Without his incredible knowledge of fluid dynamics, the clear understanding of CFD limitations would not have been realized. Dr. Jordan is a remarkable intellectual and it was an honor to receive his contributions.

I would also like to acknowledge and express appreciation to David Ferreira and Joseph Gomez, the technical staff running The University of Rhode Island's, machine shop. Thank you for sharing your years of experience to advise in the production of the parts needed to make this project a success.

Finally, I would like to thank CBC Wind Energy Solutions, Commerce Rhode Island and the University of Rhode Island for providing the research project, funding and platform to learn and advance through this unique opportunity.

TABLE OF CONTENTS

Page

Page

Page

LIST OF FIGURES

Figure **Page**

Figure **Page**

CHAPTER 1

Introduction

1.1 Overview

A traditional vertical axis wind turbine (VAWT) consists of an open rotor with fully exposed blades (Figure 1). Such designs are only able to take advantage of the wind velocity native to the environment they are placed, limiting the starting condition to greater wind speeds of which they are immersed. CBC Wind Energy Solutions aims to reduce the required wind speed needed to start a VAWT. The proposed method of doing so utilizes converging stators, surrounding the turbine, to effectively accelerate the captured wind.

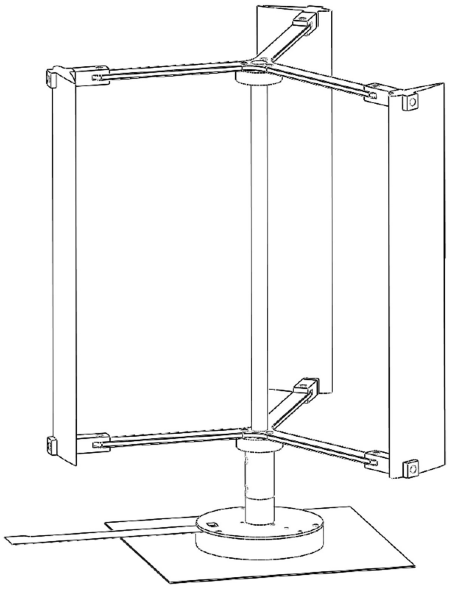


Figure 1: Traditional vertical axis wind turbine with fully exposed blades (Rolin and Porté-Agel, 2018)

1.2 Motivation

By reducing the cut in wind speed, CBC aims to capture an untapped region of the market with a Hidden in Plain Sight (HIPS) turbine. The HIPS turbine will

stand out from existing products through the utilization of surrounding stators with a narrowing cross-sectional wind capture area. The stators will accelerate the captured wind by exploiting the simple continuity equation, $A_1 v_1 = A_2 v_2$, where A is the cross sectional area of a passage and v is fluid velocity (Nelson, 2008). This assumption is valid given that the mass of air is conserved and the air is considered incompressible at the average velocities found in nature. With the ability of the HIPS design to accelerate native wind currents converging on a turbine, operation will be possible in geographic regions unsuitable for traditional designs.

A primary drawback of harnessing wind energy is the altitude at which a viable source can be attained in relation to the ground. Substantially lower average wind speeds are measured at the surface region of the velocity curve, due to no-slip boundary conditions, among other factors (Archibald, 1883). With the added obstructions of trees and houses in residential areas, useful wind energy cannot be effectively harnessed by traditional methods, without reaching well above such obstacles. This limitation can readily be seen accounted for in the implementation of existing wind energy solutions. Massive towers extend horizontal axis turbines above the tree lines or generators are placed offshore, where shear components of the wind are lowest. While effective, these solutions are too large and costly to be implemented by the average consumer.

CBC plans to create a means of capturing usable energy, while making the device available to residential and small commercial clientele, all within an aesthetically pleasing package. However, the promises of starting a turbine in conditions where other devices would fail, does not come without design challenges.

1.3 Problem Definition

Early testing of a crude HIPS prototype demonstrated some fundamental limitations. As the vertical blade tips of the central turbine rotated past the fixed

surrounding stators, an audible thumping was observed. Furthermore, as the turbine came to rest at varying orientations, relative to the stators, a stalling condition would sometimes be realized in which excessive wind speeds were required to start the turbine. These faults threatened to disrupt the benefits of the surrounding stators, prompting an initiative to study and eliminate the errors. There also existed some uncertainty encompassing the scaling requirements involved in a production unit capable of a desired power output.

1.4 Objective

CBC pursued a Commerce Rhode Island, Innovation Voucher to solve the preliminary issues and develop a fully functional prototype. This led to the involvement of the University of Rhode Island (URI), as the Innovation Voucher was awarded to the University. CBC proposed a set of prospective design parameters with the objective of selecting the optimal turbine configuration to minimize destructive fluid interferences and maximize performance. Additionally, CBC required an answer to the scaling question in order to predict prototype sizing. The scaling objective was to predict a turbine height which would produce enough starting torque to begin turning a preselected, 5 kW generator.

CHAPTER 2

Literature Review

2.1 Turbine Starting

The importance of low wind speed turbine starting has increasingly become an area of research over the past couple of decades. It is has been established that with lower cut in speed, comes the potential for greater power extraction from the wind (Wright and Wood, 2004) and (Worasinchai et al., 2012). When considering the low wind speed starting of smaller turbines, Ebert and Wood addressed the limitations of an airfoil design and the blade pitch (Ebert and Wood, 1997). It was found that the airfoil optimized for starting the turbine would perform poorly after doing so, therefore a compromise in pitch direction would typically lead to poor starting and high wind speed performance, while excelling at mid velocity conditions.

Additional research has been conducted by Zamani et al. in an effort to improve blade design to lower cut in speeds of vertical axis wind turbines (Zamani et al., 2016). No direct focus has been placed on the improvements available in surrounding stators used to accelerate the wind and improve starting abilities of VAWTs.

2.2 Surrounding Stators

Despite a lack of research focusing on the starting improvements of surrounding stators, the concept of such features has been explored by (Burlando et al., 2015). The study correlated experimental wind turbine results, using stators, to computational fluid dynamics (CFD) models. The study evaluated a few performance improvements for the dynamic state of the turbine, cautioning of real-world variability due to wind instabilities. However, the investigation did

offer some optimistic results to encourage the use of such stators. With a scarcity of wind turbine experiments including surround stators, this research aims to determine if stators benefit VAWT starting.

2.3 Wind Turbine Scaling

Previous work has been conducted by Giahi and Dehkordi on the efficacy of Similarity Theory to form equations which predict the output of scaled CFD models (Giahi and Jafarian Dehkordi, 2016). The research showed agreement by numerical processes to the scaling equations, proving an increase in torque output occurs by a relation to the cube of the rotor diameter. This work demonstrated the ability to effectively scale torque data of horizontal axis wind turbines (HAWT). In this research a variant will be investigated, given the focus of a vertical axis wind turbine and scaling parameters relating to wind velocity and output torque.

2.4 Design Optimization

With a vast increase in the interest into renewable energy sources, wind energy has become a highly researched area of focus in later years. Much analysis exists on the characterization of design performance for traditional turbine configurations. Complex designs aim to advance performance of large scale wind turbines, such as the variable pitch design proposed by (Jafarnejadsani et al., 2013) and the deformable blade design proposed by (Alejandro Franco et al., 2015). These systems are cost effective to build into enormous commercial units to allow adaptive function from cut in, through all wind speeds. However, these solutions are not viable for smaller residential turbines, where the advanced mechanisms and control systems will drastically lower the return on investment.

2.5 Wind Turbine Power Output

The primary focus, when considering a wind turbine design, is the efficiency by which it can extract energy from what is available in a given volume of wind. A determining limit to this efficiency is reviewed by (De Lellis et al., 2018), known as the Betz limit. The rational concludes that a maximum of $16/27 \approx 59\%$ of the wind's energy can be harnessed by a horizontal axis wind turbine. Compare this with an average efficiency, of the leading turbine designs, between 35-45%, without considering losses due to mechanical systems such as generators as well as power transmission and conversion.

However, this theoretical cap does not discourage the innovation involved in trying to meet or exceed this figure (Franković and Vrsalović, 2001). Some propose that the limit is not an exact value and it may be possible to exceed the limit by a small fraction (Farthing, 2013). However, it is agreed the Betz limit is a very accurate approximation and should be considered during wind turbine design.

CHAPTER 3

Test Plan

3.1 Problem Development

Through working with the University of Rhode Island a final turbine design was reached by CBC. The design incorporated eight surrounding stators, in order to accommodate a greater variation in wind direction, while producing consistent channeling. During prior testing of a crude plywood prototype, observations were made of a thumping sound produced as the turbine blade tips would pass the interior edge of a stator. Depending where the turbine came to rest during wind lulls, different wind velocities would be required to start the turbine. It was found that some turbine to stator offset angles would require large currents to re-start the motion.

Some final revisions were proposed which would account for the stalling condition along with the interference between stators and turbine. A hinged, or open-stator design was imagined, along with the addition of a helical turbine. The theory behind the open stator design was that while close proximity of the stator to turbine aided with starting, at higher wind speeds, the sealing of the windward cavity would bottleneck airflow. Therefore, allowing the stators to move away from the turbine would alleviate this blockage. Additionally, a helical turbine was expected to reduce the area of passing blade and stator to a single point, rather than an entire edge. It was speculated that a reduction of symmetry from the top to the bottom of the turbine may also contribute to a reduction in stalling.

3.2 Solution approach

The approach to understanding the benefits of the final design elements involved scale model testing of replica turbines with configurable geometries. A

replica with a 5-inch-tall turbine blade would be designed and built to fit within the test chamber of URI's Aerolabs wind tunnel. A number of revisions led to the design of a turbine model allowing for the interchange of straight stators and open stators. The model would also require the exchange of turbines with varied helix angles.

With consideration to a number of performance variables, the 5-inch replica would accommodate multiple sensors in conjunction with the combination of open or closed stators, paired with varied turbine helices. Testing would be conducted to quantify the static torque production for a range of starting angles of the turbine in relation to the stators. A selection would then be made of the top performing helix angle for comparison to the straight turbine in the remaining tests. Turbine acceleration in a constant flux would be measured for the two turbine cases combined with open and closed stators. The same combination would then be applied to a variable wind speed while measuring angular velocity. Finally, the best performing combination would be retested for static torque production in an incrementally increasing, but constant wind speed condition, with many data points captured and averaged.

This final data would be scaled using non-dimensional analysis, to predict the turbine sizing required to start a 5 kW generator. The scaling method would then be validated by building a 3D printed, 2.5-inch turbine replica and performing the same final torque analysis. If torque data could be reliably scaled between models to predict the opposing behavior, the method would then be employed to determine a full-size prototype. With construction complete of the 5 kW prototype, static testing would be conducted once more, on the full size unit, to validate the prediction.

CHAPTER 4

Design

4.1 Mechanical Components

The success of the replica turbines hinged largely on the design and preparation of the mechanical system, with parts selection, fabrication techniques and accuracy requiring great consideration. A number of manufacturing design iterations were developed before arriving at the selected methods. The steps and requirements are outlined in this chapter.

4.1.1 Requirements

These requirements would guide the material selections and sensor items used in the model. A list of design requirements for the physical turbine mechanism follows:

- Fabrication choices must not compromise the exactness of the replica to the original model design.
- At least one surface of the turbine top or bottom housing must be transparent to visualize turbine alignment.
- A minimalistic stator fastening approach is required to avoid adding turbulence to the system.
- The fastening method needs to accommodate both straight stators as well as angled stators.
- The turbine assembly must be indexable about the central axis to change headwind angles of the windward stator.
- The turbine assembly should be supported with enough rigidity to minimize harmonic oscillations.
- The turbine assembly should ride on precision bearings and shafts to mini-

mize vibration and frictional forces.

- The output shaft and upper and lower support plates must be able to accommodate a load cell or an optical rotary encoder.

4.1.2 Parts Selection - 5-inch Model

All design iterations were modeled using SolidWorks to confirm part geometries and fitment of assembly components. An exploded view of the model can be seen in Figure 2. Each part, its material, and design consideration will be discussed in this chapter.

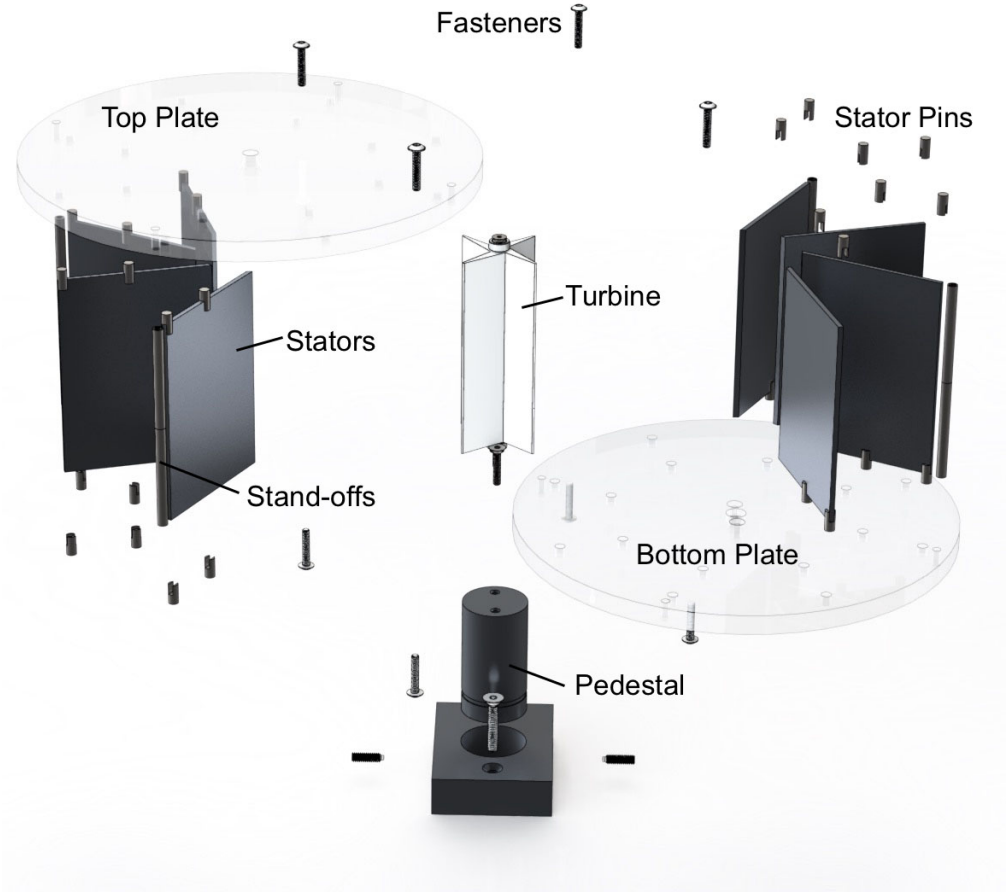
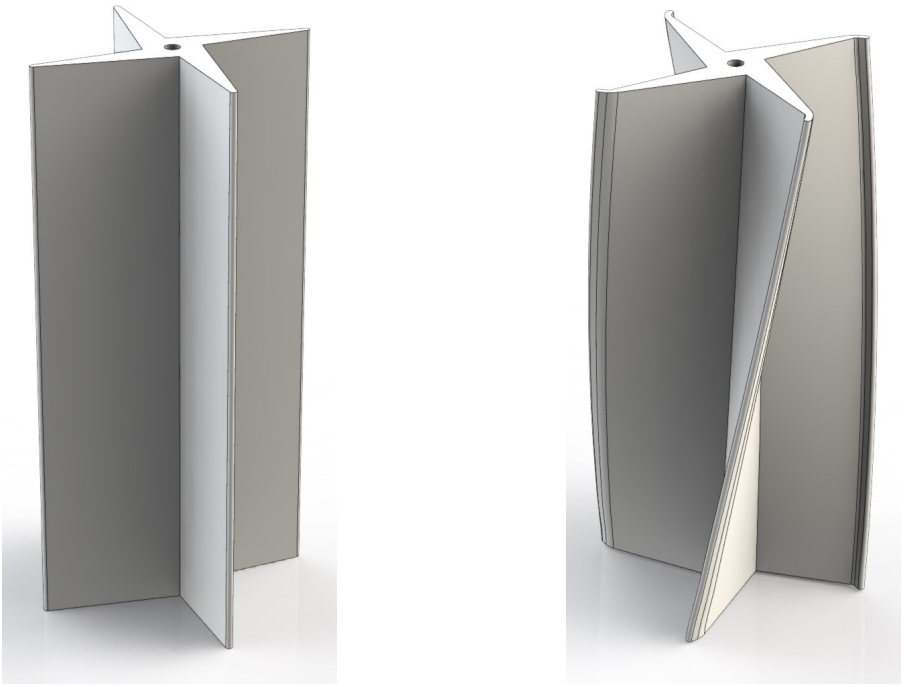


Figure 2: SolidWorks rendering of the 5 inch, revision 1 replica with exploded parts view

To minimize fabrication requirements, off-the-shelf components were incorporated into the model design wherever possible. 0.5-inch transparent polycarbonate was selected for the top and bottom surface plates to allow for an unobstructed view of the stators and turbine. For the stators, 0.125-inch aluminum sheet was chosen. The rigidity of the aluminum allows for minimal fastening points along the top and bottom edges, giving freedom to include variable stator angles without constriction. With a ridged stator design, a open-angle was to be set by notching and bending the stators. The primary test angle of 15° was chosen by the CBC team. Stainless steel rod was chosen to fabricate support pins to fasten the stators between the two plates. These pins were notched to grasp the stators at four, in-plane points behind the bending region of the stator.

3D printing techniques were selected to achieve the complex geometry and fine detail needed for the turbines, which ride on precession bearings supported by slip-fit bores on the top and bottom plates. The turbines were also outfitted with press-fit bores through the center, where a 6 mm shaft is securely attached. A variable helix angle was designed into three turbines. The angle is defined by the offset of the top and bottom surface, a 0° and 45° offset can be seen in Figure 3a and 3b, respectively. Joining the two plates, standoffs maintain the turbine height and hold the assembly fixed. The design allows for the interchange of stators and turbine with the removal of only four screws.



(a) (b)

Figure 3: Zero and 45 degree offset turbine design renderings

The entire unit was originally supported by an indexable, machined aluminum pedestal. The design positioned the unit directly in the center of the 12 in x 12 in cross section of the wind tunnel testing chamber. However, harmonic oscillations developed as flow enveloped the turbine. Alternatively, a polycarbonate mounting plate was designed to mount the turbine directly to the base of the test chamber. The assembled rendering of the 5-inch replica can be seen in Figure 4.

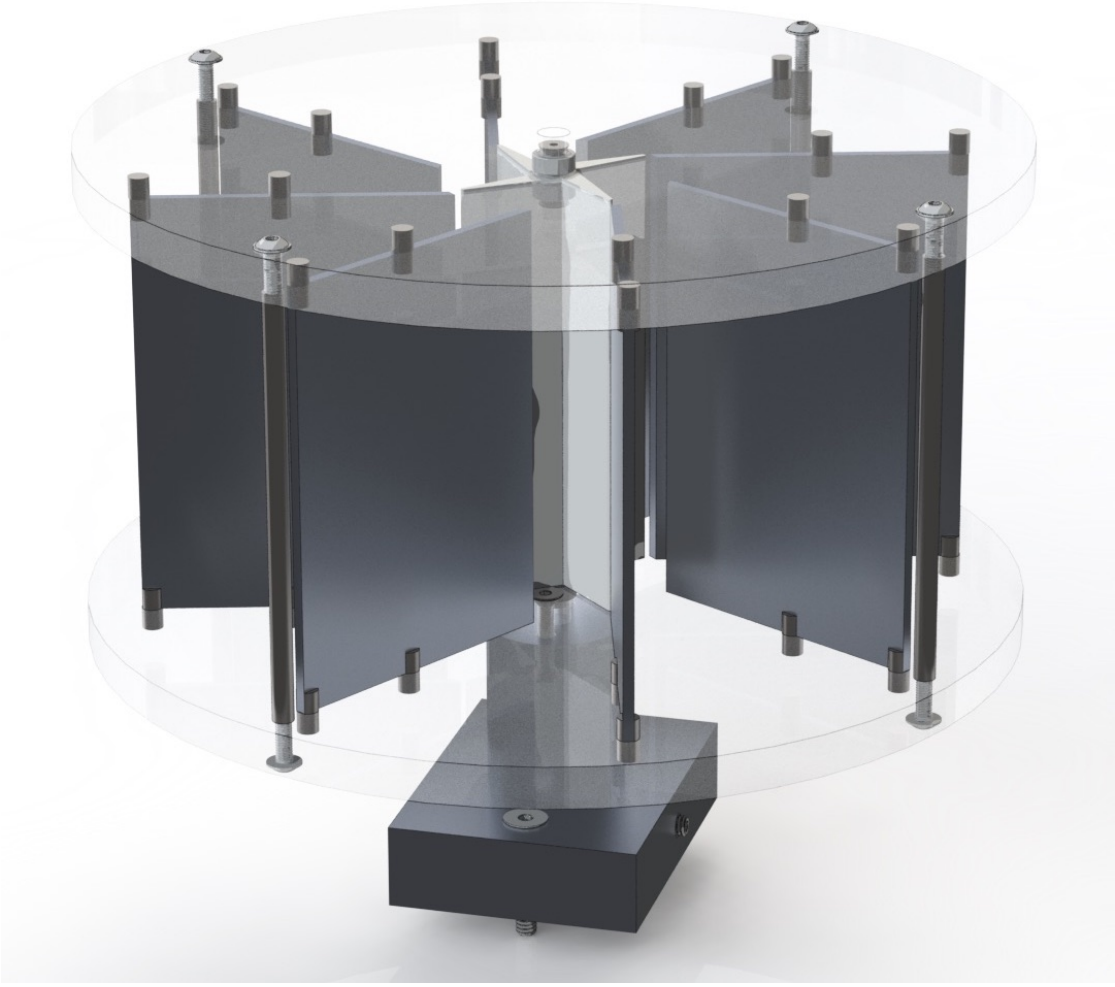


Figure 4: SolidWorks rendering of the 5-inch, revision 1 replica. Front/top view

4.1.3 Parts Selection - 2.5-inch Model

Following primary testing and optimal component selection, the 2.5-inch replica was designed in a single configuration solely to prove scaling predictions (Figure 5). Therefore, it was desirable to interface with the same data acquisition hardware as the 5-inch model. This meant the same 6 mm output shaft would be used and mounting will be possible to the same base plate. In order to make use of the same 6 mm output shaft, the central turbine column of the 2.5-inch replica had to be enlarged slightly. This meant the model would not be a precise

scale replica of the 5-inch version, however, this deviation was accounted for in the scaling equations.

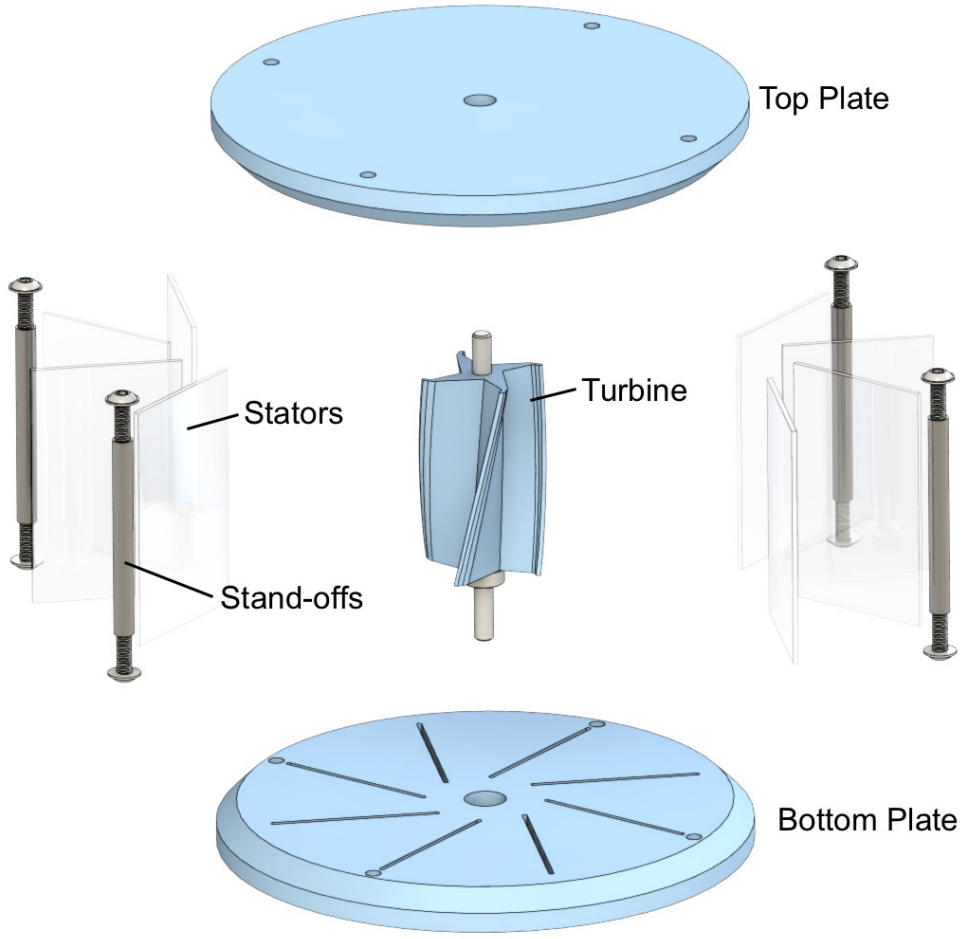


Figure 5: SolidWorks rendering of the 2.5-inch, replica. Exploded parts view

Given the fine detail of the smaller unit, an almost entirely 3D printed model was designed. A Stratasys Objet printer was selected to create the top and bottom plates, along with the turbine. The same 0.25-inch standoffs were selected to hold the device together and the top and bottom plates were designed with slots to capture straight 0.0625-inch polycarbonate stators. The final assembled design rendering can be seen in Figure 6.

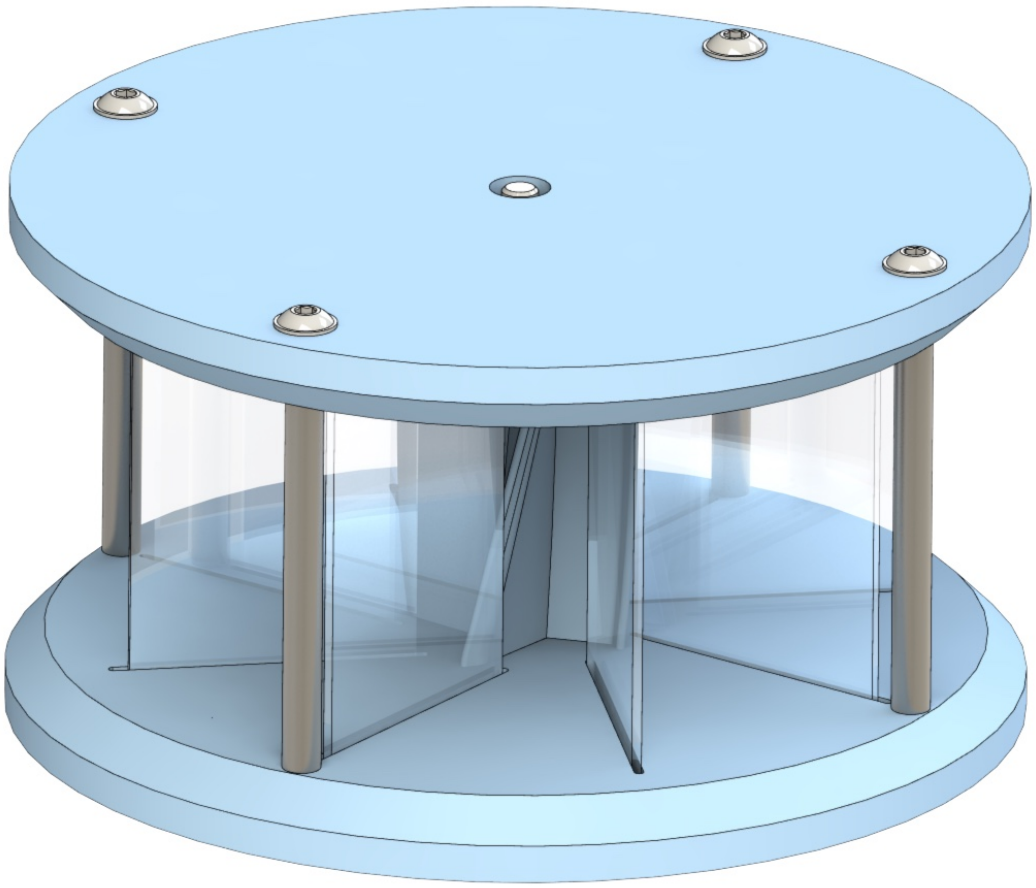


Figure 6: SolidWorks rendering of the 2.5-inch, revision 1 replica. Front/top view

4.2 Electronics

Of equal importance to the mechanical design of the replica turbines, is the electronic interface needed for data acquisition. Precise model integration and user control is needed for the delicate parts and minute changes in state. As with the turbine mechanisms, a number of design revisions led to the final electronics setup, which is detailed in this section.

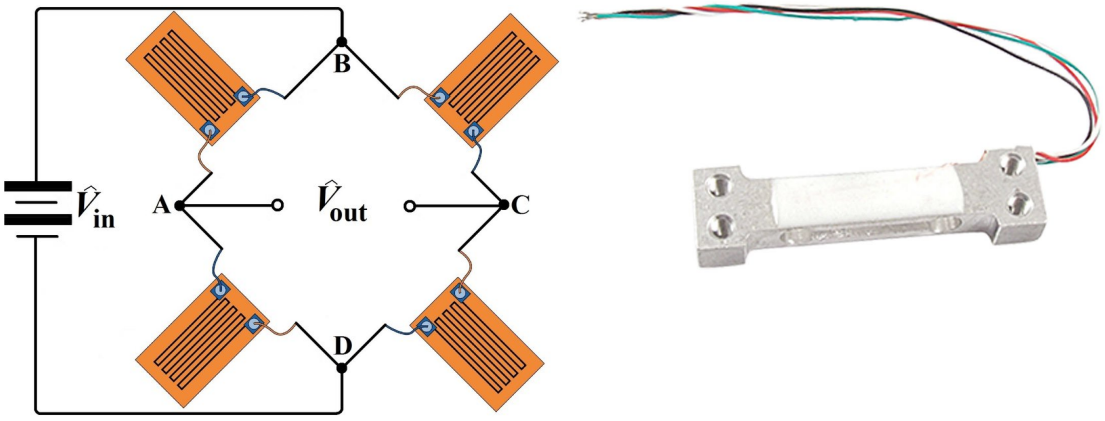
4.2.1 Requirements

Following the physical requirements of the turbine model, specific measurements were to be recorded during testing. The requirements pertaining to the data acquisition components were as follows:

- Static torque measurements must be captured while the turbine is securely held in place.
- Torque measurements must be read from 0 to 0.018 Nm with accuracy to 1e-6 Nm.
- Angular position must be captured every 0.3 degrees.
- Wind velocity must be captured in a range of 0 to 35 m/s in divisions of, at most, 0.01 m/s.
- A capture rate down to 100 Hz must be achievable.
- All items must be configurable as a portable, weatherproof, stand alone system for outdoor use.

4.2.2 Component selection

With a selected lever arm length of 0.0193 m and a 100 g capacity load cell, the unit was effectively able to measure $0.98N \times 0.0193m = 0.019Nm$, exceeding the 0.018 Nm requirement. The load cell uses four strain gauges in a full Wheatstone-bridge configuration (Figure 7), providing a linear relation between loading and voltage output. To generate an excitation voltage and amplify the output values from the strain gauges, a Texas instruments INA125P, 16-pin, dual inline package, integrated circuit, was used. This allowed for the use of a 5 V reference, with a low noise 1 kHz signal. To capture the analog voltage signal with adequate resolution, a 16-bit analog to digital converter (ADC) was used, giving 65,536 voltage divisions. The Adafruit ADS1115 (Figure 8), 4 channel ADC was selected for its high precision and 860 samples/second capabilities.



(a) Full Wheatstone bridge diagrams with 4 strain gauge configuration

(b) Ucell, four strain-gauge, 100 gram load cell

Figure 7: Strain-gauge diagram and part

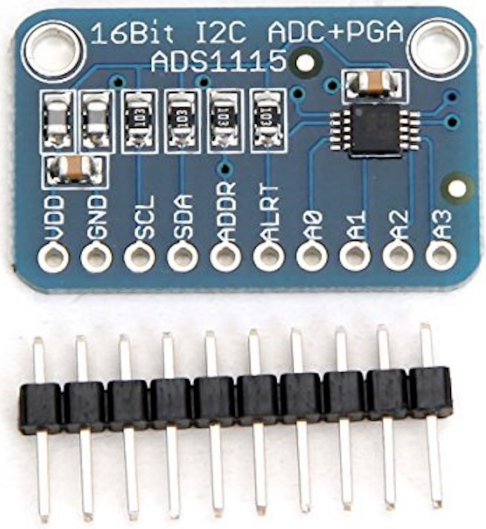
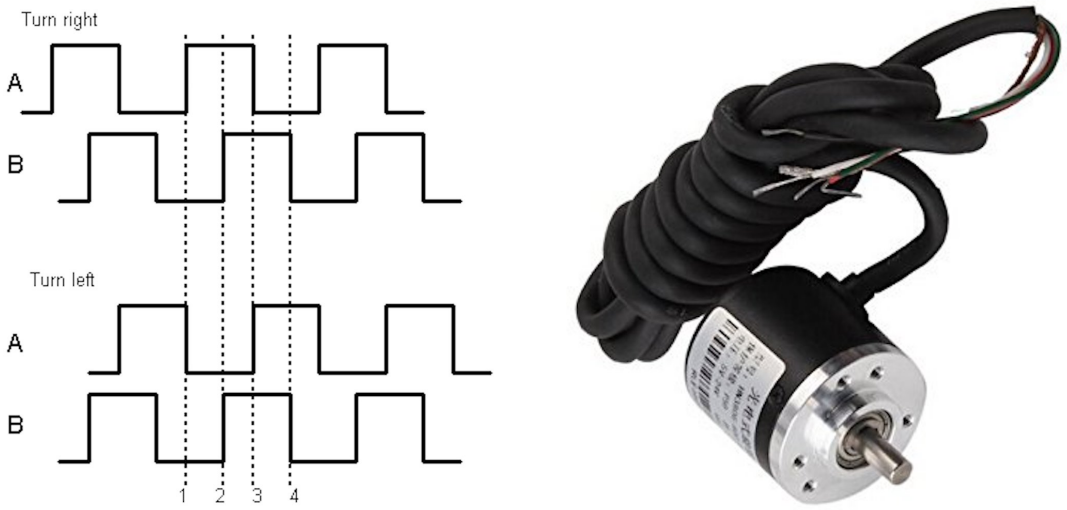


Figure 8: Adafruit 16 bit, 4 channel analog to digital converter

With the load cell, amplifier and ADC selected, an Arduino UNO could be used to interface with the sensors. These devices enable the capture of static torque measurements, but will not take dynamic readings. A 600 pulse per revolution

(ppr), glass rotary encoder was added to gather angular displacement data. With the quadrature encoding configuration (Figure 9), an interrupt from the rising and falling edge of both channels could capture 2400 total ppr, exceeding the design requirements.



(a) Rotary encoder phase diagram for quadrature output

(b) Sigwise 600 ppr rotary encoder with 6 mm shaft

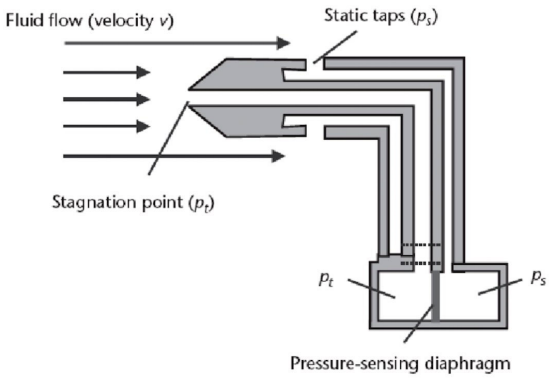
Figure 9: Quadrature encoder and diagram

The flow of air is the remaining parameter to capture. For wind tunnel readings, this was achieved using a differential pressure sensor made by Sensirion (Figure 10a). The device is capable of measuring a pressure range of -0.02kPa to 0.5kPa with a 0.1% error and 40 ms response time. When coupled with a pitot static tube (Figure 10b), we can consider Bernoulli's equation of static pressure + dynamic pressure = total pressure and rewrite it to solve for stream velocity (Equation 1).

$$V = \sqrt{\frac{2 * (p_t - p_s)}{\rho}} \tag{1}$$

where p_s is the static pressure perpendicular to flow, p_t is the stream pressure measured in the direction of flow, V is fluid velocity and ρ is fluid density.

(a) Sensirion SDP1000 differential pressure sensor



(b) Static pitot tube cross section diagram

Figure 10: Pressure sensor and static pitot diagram

Finally, outdoor wind currents were measured using Adafruit's analog voltage anemometer (Figure 11). The device is capable of recording wind velocities up to 32.4 m/s over a 1.6 V span. All devices can be coupled with the Arduino microcontroller with data capture controlled via software.

Figure 11: Adafruit anemometer wind speed sensor with analog voltage output

4.3 Software

The electronic hardware components are only as good as the readings that can be recorded from them. Therefore, custom software was written for the Arduino microcontroller and client tool to interface with the device.

4.3.1 Design Requirements

As with the mechanics and electronic components, key requirements needed to be satisfied by the acquisition software. Given that the software was developed specifically for the purpose of the aforementioned tests, versatility was not an issue. The primary requirements set forth for the software are as follow:

- The microcontroller should operate off precise timing interrupts for clock data.
- Interrupts must be triggered by the encoder so no rotational data is lost.
- A capture rate down to 100 Hz must be achievable by the microcontroller.
- Transmission protocol between microcontroller and client software must support a bitrate able to transfer time, pressure, force and rotational data at 100 Hz.
- A graphic user interface should support real-time data capture of air pressure, force reading and angular position, as well as user control of calibration and recording.
- Large data sets containing tens of thousands of points must be processed quickly and automatically while allowing for data to be operated on.
- A portable version of the software must be deployable from an SD card, in the test field.

The stripped-down C++ language run by the Arduino UNO interpreter will provide the necessary capabilities for capturing data from all hardware sources, with strategic programming. Having an achievable baud rate of 230400, the Ar-

duino is able to receive commands and transmit data over a serial interface, without bottle-necking data, to maintain real-time capture. For the software used to interface with the Arduino, Visual Basic was chosen. Given its ability to communicate with the microcontroller over a serial interface, while displaying the real-time data and accepting user input from button presses, this choice proves adequate.

Finally, with mass amounts of captured data, MatLab would be used to handle and process the information. With data structuring capabilities, the script could store and pole data in a user-friendly manner. Automation was also built into the scripts to import and handle data.

CHAPTER 5

Build

5.1 Mechanical Components

For the components of the turbines that could not be sourced off-the-shelf, custom parts were made. A number of core processes were involved in the roughing out of parts. These included CNC routing, hand milling and turning, along with 3D printing.

5.1.1 5-inch Turbine Replica

The top and bottom plates were CNC machined using a Laguna IQ CNC routing table (Figure 12). Each plate was outfitted with holes for mounting hardware as well as the bores for 16 stator positioning pins, on each. The half-inch thick polycarbonate was routed to 11 inches in diameter, with 0.25-inch pin-bores halfway through the material and a 0.5-inch bearing seat in the center (Figure 13). The top and bottom plates are mirror images, and the attachment panel is mated through the pedestal mounting holes.

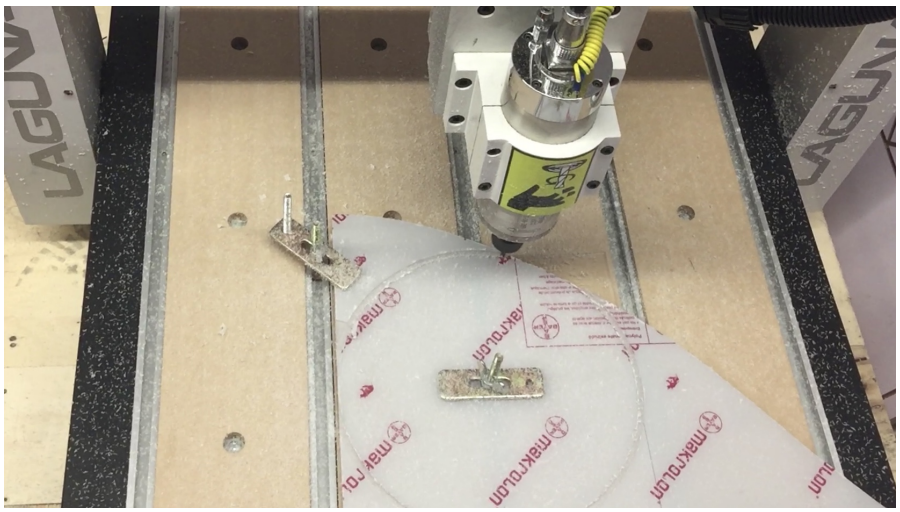


Figure 12: Machining the mounting plate using a Laguna IQ CNC router

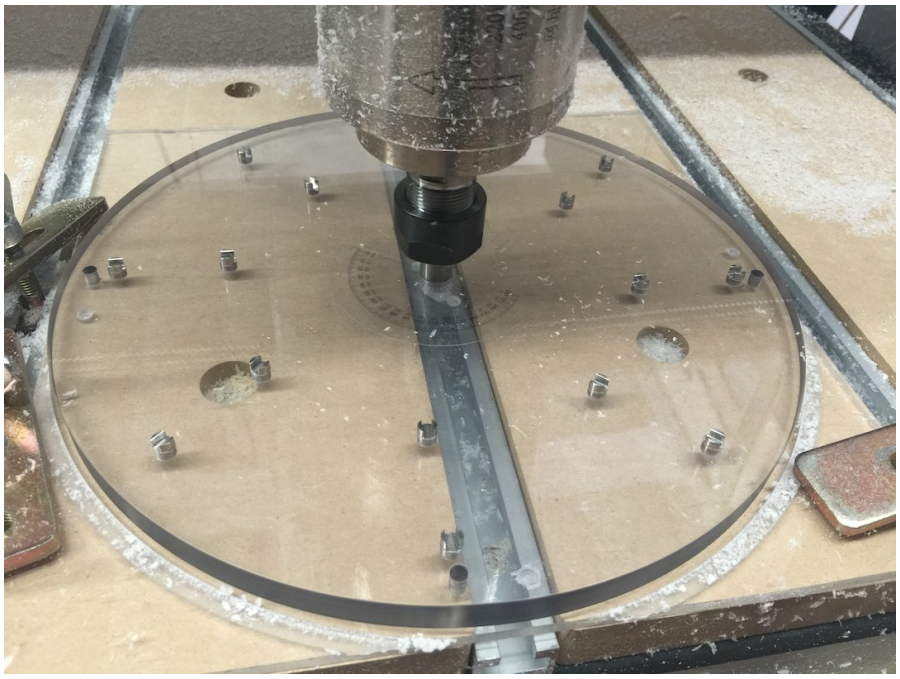


Figure 13: The completed 5-inch turbine top plate, with stator pins installed

The 0.125-inch aluminum sheet for the stators was cut to rough dimension using a shear. The parts were then clamped together in a vise and brought to final dimension using an endmill. At this time, a 30° bevel was cut into the stator edge closest to the turbine. A set of eight stators was left flat, while another eight were notched in the center, along the length, with an 0.125-inch endmill. The cut allowed for a precise bend to the 15-degree, open position. Finally, the surface was sand blasted with a fine grit media to produce a consistent surface finish between all stator parts (Figure 14).

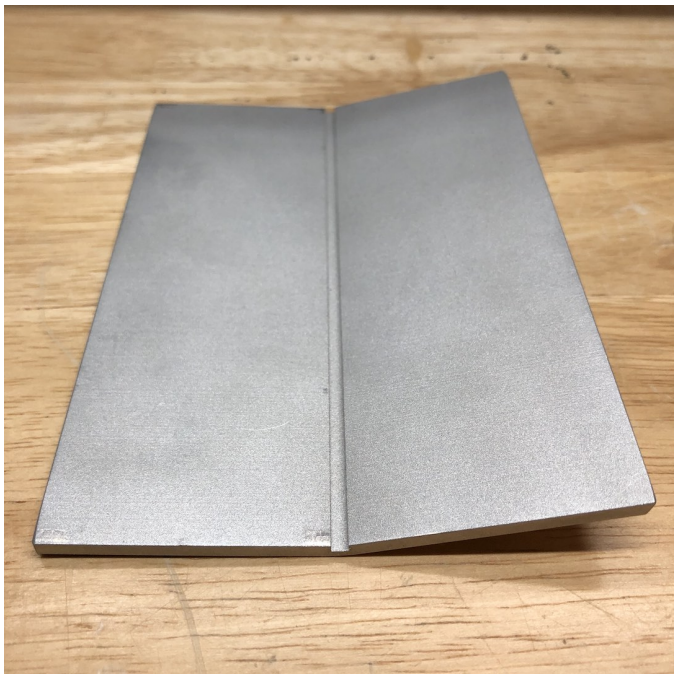


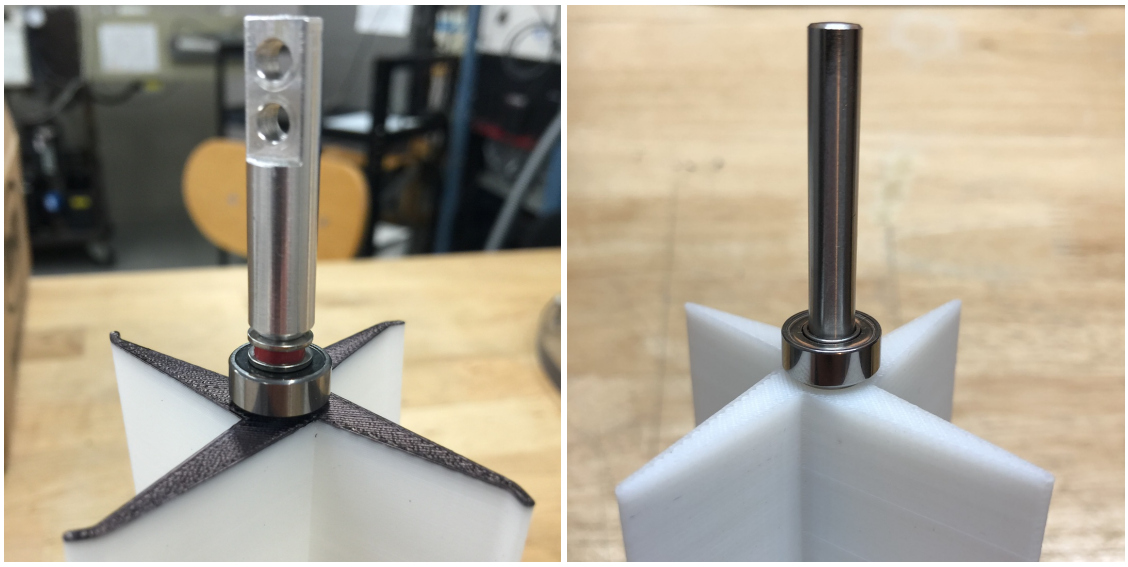
Figure 14: A 15° stator machined from 0.125-inch aluminum

Attaching the stators to the top and bottom plates are notched locating pins. The pins began as a length of 0.25-inch stainless steel rod, which was chamfered and cut to exact 0.5-inch lengths using an EMCO lathe. The pins were then individually mounted in a collet block and slotted using a 0.125-inch endmill (Figure 15a), just larger than the plate thickness. The pines were finished by strip sanding the hard edges to ease installation and changeover of parts (Figure 15b).

(a) (b)

Figure 15: Stator mounting pins slotted form 0.25-inch stainless steel rod

Three turbines were 3D printed with solid fill ABS plastic, using a Stratasys Dimension BST 768 printer. A straight (0°), 15° and 45° version was printed. The first revision accepted an 8-32 bolt on either end (Figure 16a), which output shafts were machined to fit. However, the non-continuous shaft led to balance issues, leading to a solid shaft revision (Figure 16b). For the second revision, a central bore was printed at its nominal size to provide a medium drive, force-fit. The 6 mm shaft was installed using an arbor press, locking the shaft and turbine together. Two of the completed turbines are seen in Figure 17, which are the straight and 45° offset version, from left to right.



(a) (b)

Figure 16: Turbine output shaft integration methods. Threaded insert (a) and continuous shaft (b)

Figure 17: Straight and 45° offset turbines with fitted shafts and bearings

The indexable pedestal was machined from a solid rod and block of aluminum (Figure 18). Drilled and tapped for mounting hardware and nylon tipped set

screws, to fix rotation and mount the base to the matched bolt pattern found in the base of the wind tunnel. However, mounting about the center of the turbine base plate restricted access to the lower output shaft, requiring force measurements to be taken from inside the wind tunnel. Resonance from turbulent wind flow over the unit led to unstable readings from the load cell, along with harmonic oscillations of the entire unit. The design was revised to eliminate the pedestal, mounting the turbine directly to a custom wind tunnel floor plate (Figure 19). This also allowed for the output shaft to pass through the bottom of the wind tunnel for instrument fixtures separate from fluid flow.

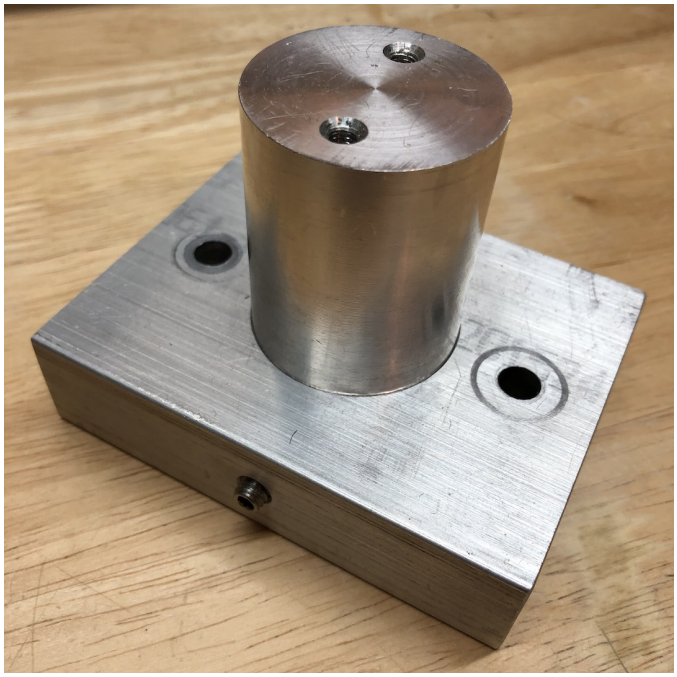


Figure 18: Indexable turbine pedestal used to mount the replica to the base of the wind tunnel

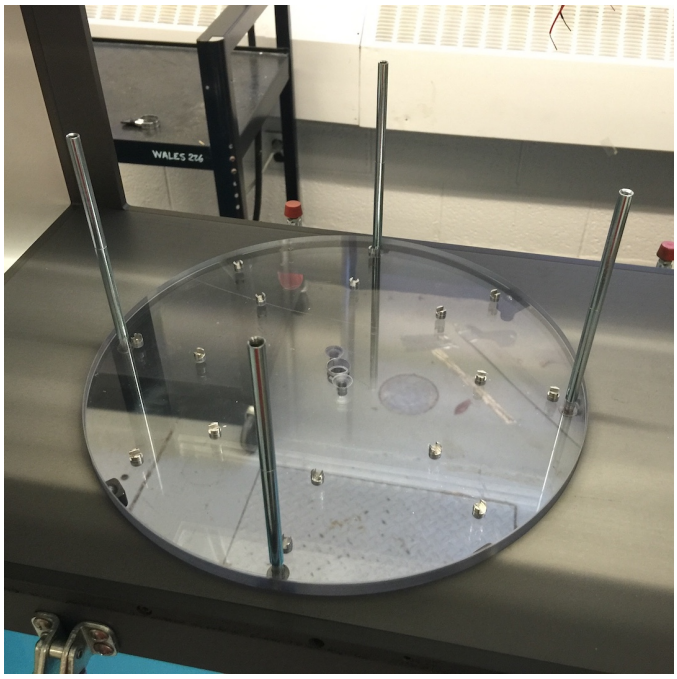
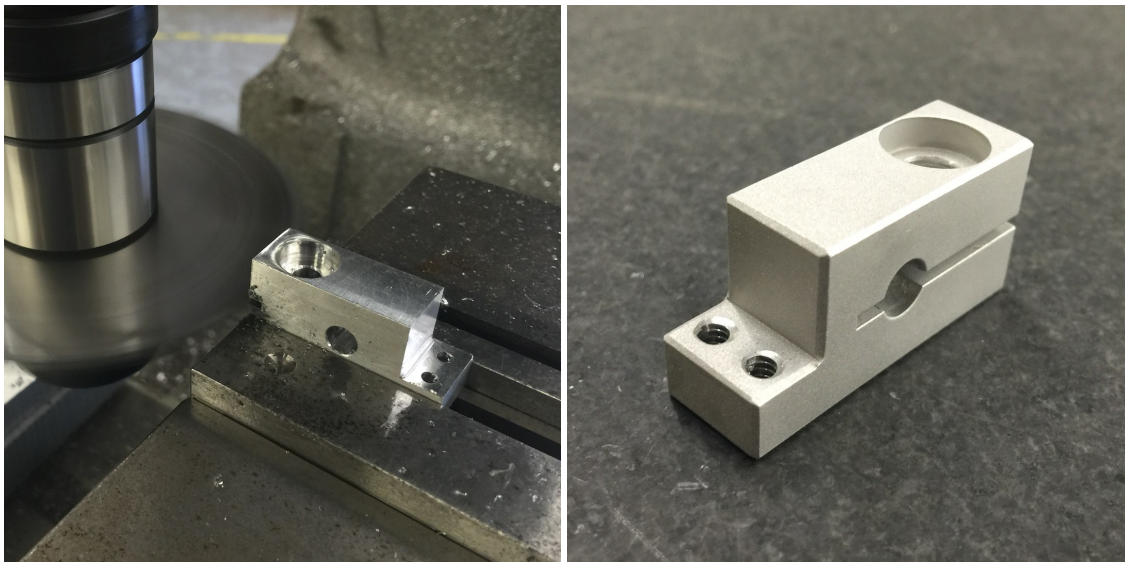


Figure 19: Bottom plate and mounting plate attached and installed in the Aerolabs wind tunnel

To measure the static output torque of the turbine, a mounting bracket was milled from a solid block of aluminum. The lever arm was offset from the load-cell contact point to the center of the output shaft by exactly 0.0193 m. The adapter was slot cut (Figure 20a) to allow the lever arm to clamp the output shaft in different orientations. This feature would allow for the quick release and positioning of the turbine and shaft. The finished mount is shown in Figure 20b.



(a) (b)

Figure 20: Load cell mounting bracket for attachment to output shaft. Slot machining (a) and complete (b)

With all the parts complete, the turbine could be assembled. A base configuration of straight stators and straight turbine can be seen in Figure 21, of the first revison, 5-inch replica.

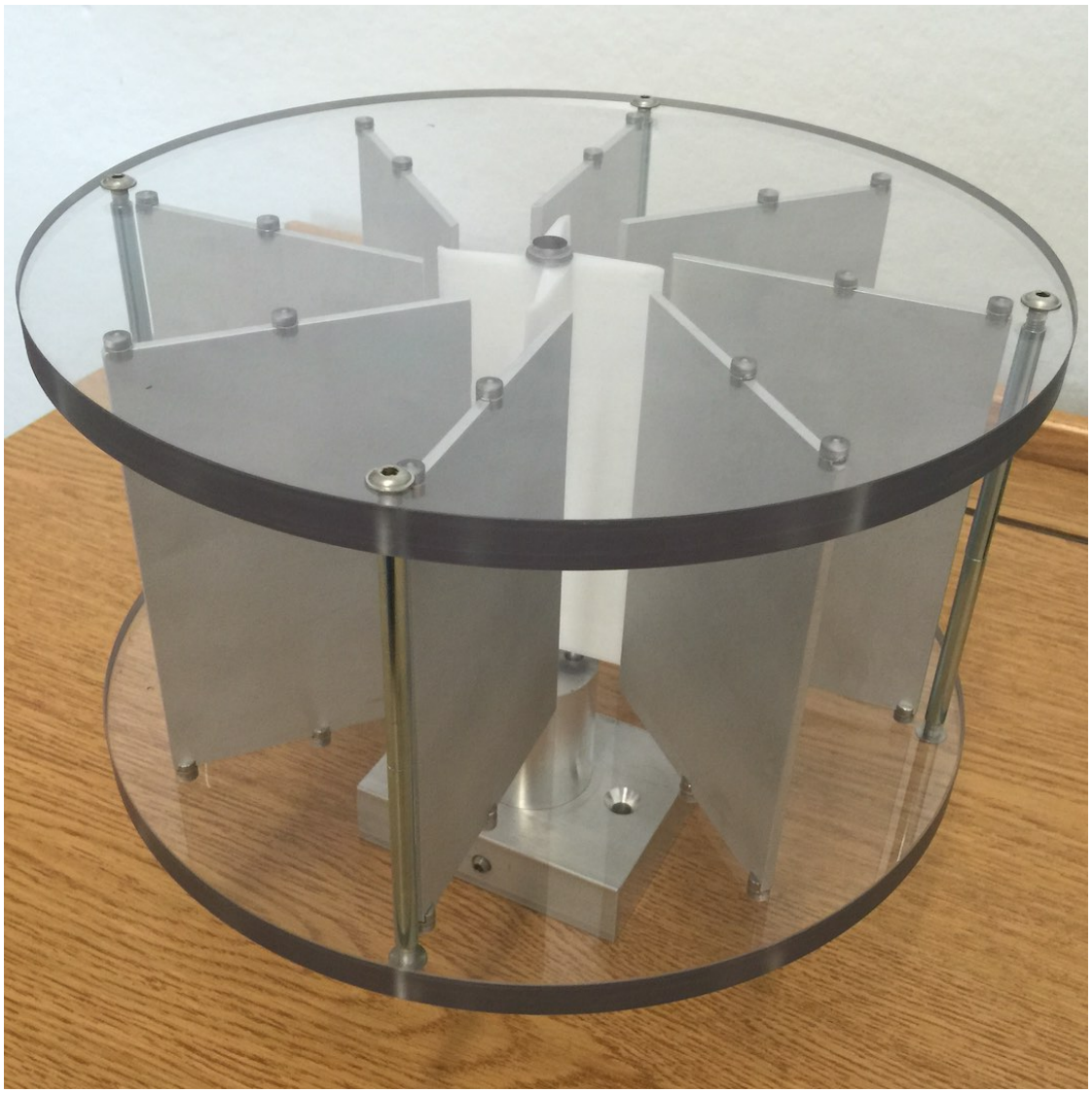
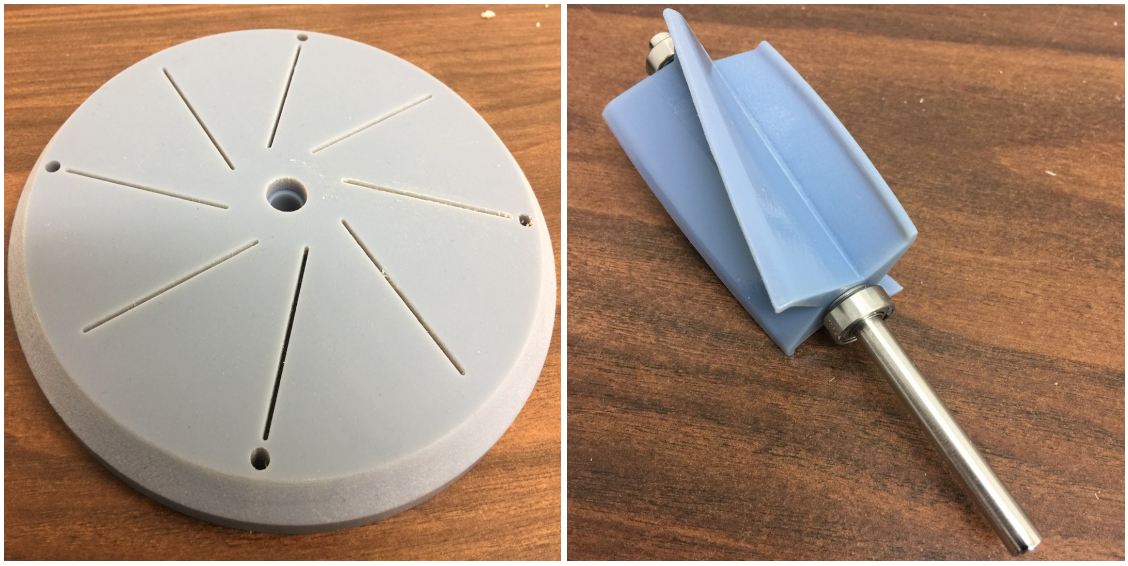


Figure 21: 5-inch replica model shown with straight stators and straight turbine

5.1.2 2.5-inch Turbine Replica

The design of the 2.5-inch replica interfaced with the same baseplate and load cell mount as the 5-inch model, reducing the number of duplicate parts needed. A Stratasys Objet 30 3D printer was used to produce the base plate and turbine from Vero opaque photopolymers (Figure 22). The high-resolution printer provided a smooth finish and extremely accurate parts helping to reduce surface eddies. The

assembled 2.5-inch replica can be seen in Figure 23.



(a) (b)

Figure 22: 3D printed components of the 2.5-inch replica. (a) Base plate and (b) turbine assembly

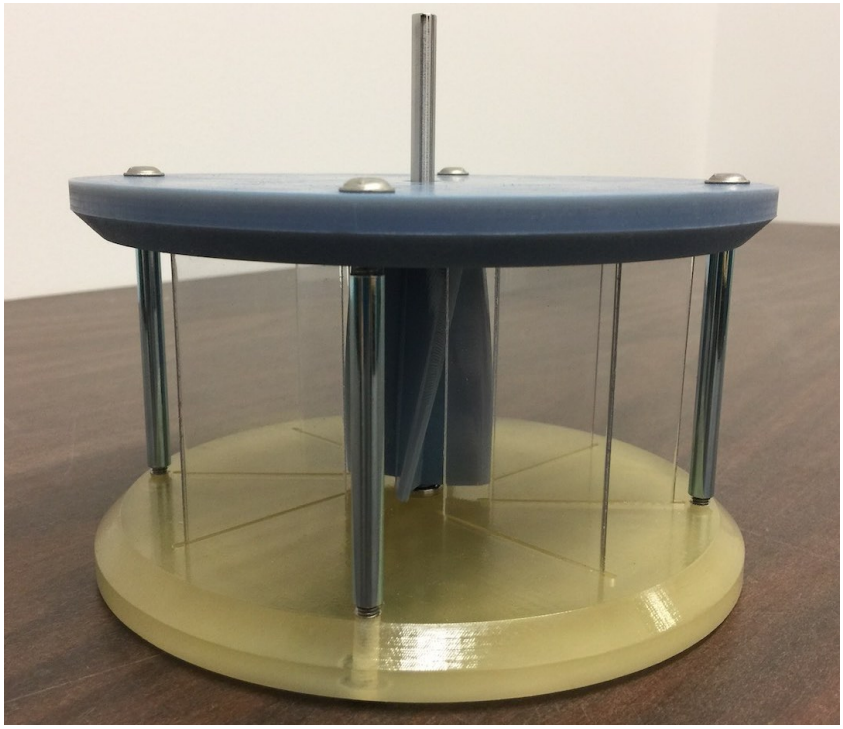


Figure 23: 2.5-inch replica model shown with straight stators and 45° turbine

5.1.3 7-Foot Turbine Prototype

The entire build of the 7-foot prototype was carried out by private contractors arranged by the CBC team. However, data acquisition was not accounted for. A 50 kg load cell mount was fabricated from 0.25-inch aluminum plate. The load cell was supported on an upright, perpendicular to the base, fastened to a length of extruded 80/20 aluminum stock. The unit (Figure 24) was mounted to the base of the turbine prototype using industrial double-sided tape. To capture the rotation of the turbine, a 6 mm flex coupler was bonded to the generator shaft, which rotates 1:1 with the turbine. The encoder was then attached to the coupler using a set screw. The assembled 7-foot prototype is shown in Figure 25.

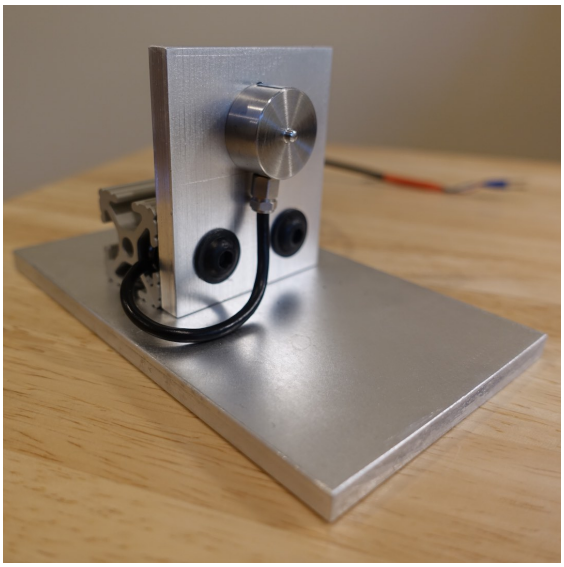


Figure 24: 50 kg load cell mount for interface with 7-foot prototype

Figure 25: 7-foot prototype with opening stators and 45° turbine on the URI Bay Campus

5.2 Electronics

Numerous steps were taken in preparation for the assembly of the electronic components. The circuits were laid out virtually before ordering or testing parts. However, some originally sourced parts did not meet design criteria upon testing, leading to the final build revision outlined in this section.

5.2.1 Replica Models

All electronic components were standard between the 2.5-inch and 5-inch replicas. The circuit was first designed using software to represent the breadboard layout (Figure 26). The schematic was generated from the breadboard layout (Figure 27). The design was then implemented using the selected components joined by jumper wires on the breadboard layout. The load cell amplifier was separated from the data acquisition circuit to keep it as close to the strain gauges as possible. Long transmission wires were found to impede the signal and compromise results by adding noise. The assembled Arduino components can be seen in Figure 28 and the amplifier circuit in Figure 29.

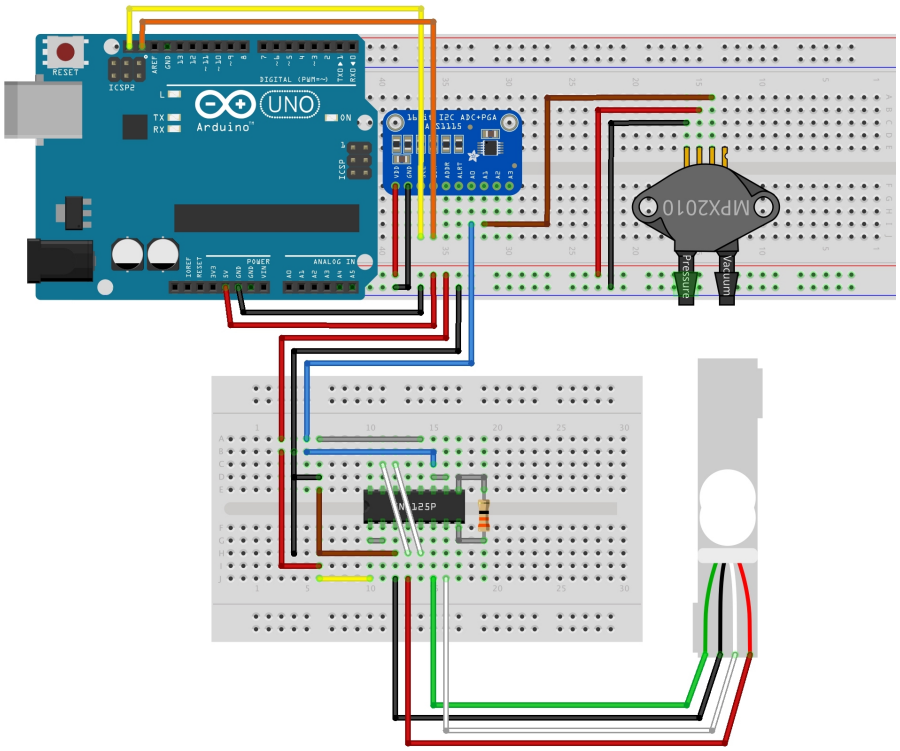


Figure 26: Breadboard layout of electronics to capture torque, angular velocity and wind speed

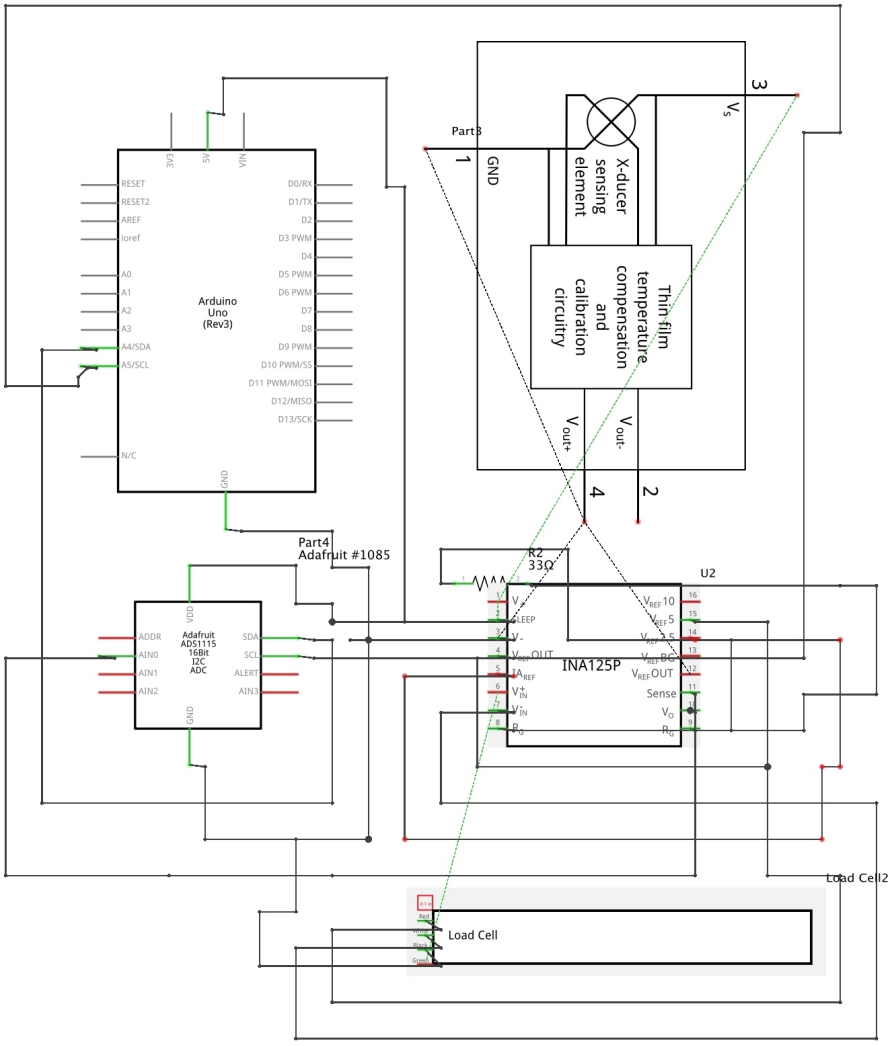


Figure 27: Wiring diagram of electronics to capture torque, angular velocity and wind speed

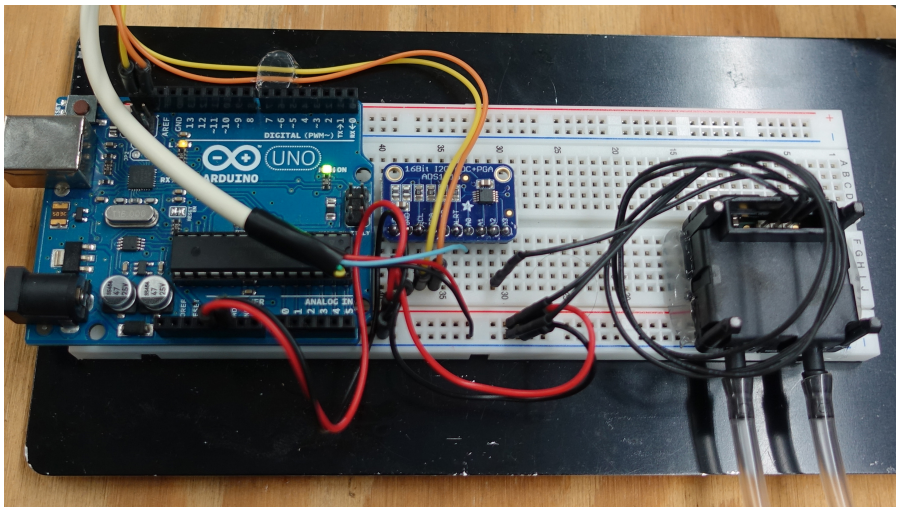


Figure 28: Arduino microcontroller coupled with 16 bit ADC and differential pressure sensor

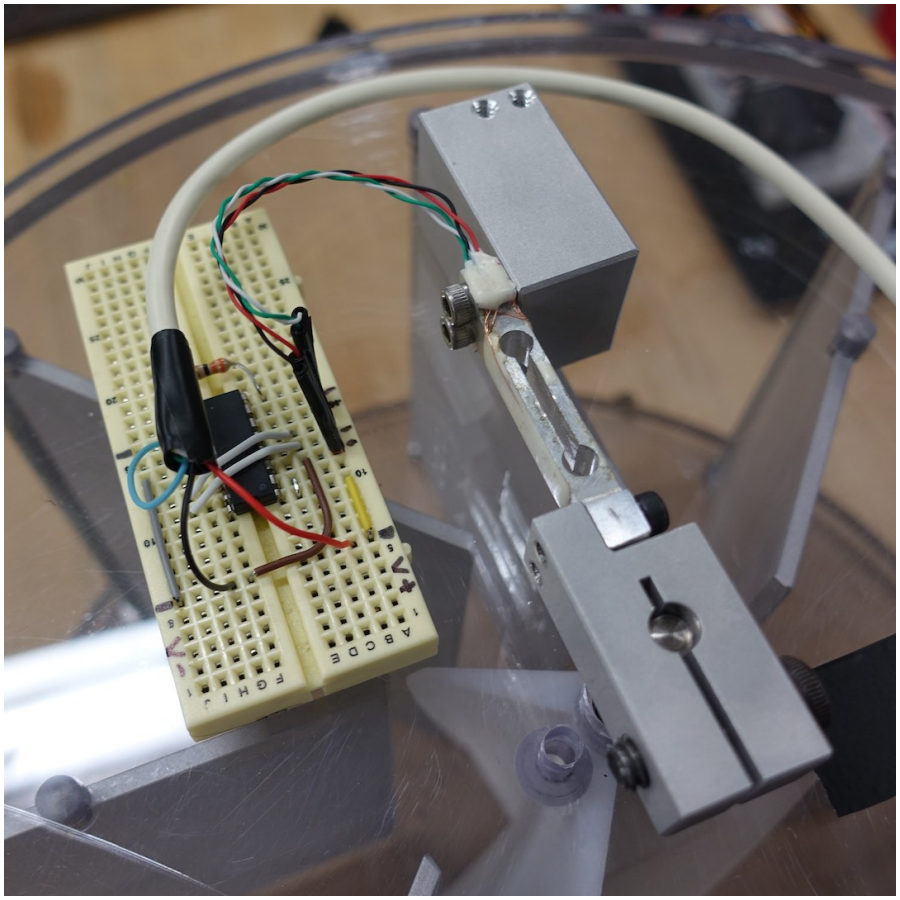


Figure 29: Load cell amplifier paired with 100 g load cell

The encoder, used for monitoring angular displacement, could be exchanged for the load cell from outside the wind tunnel. The encoder (Figure 30) made use of a 6 mm coupler to join it to the output shaft and locate it concentrically.

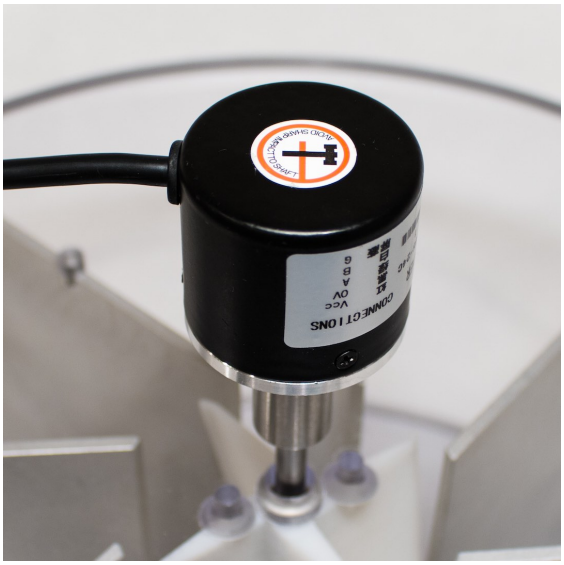


Figure 30: 600 ppr glass encoder mounted on the 5-inch turbine output shaft

5.2.2 7-foot Turbine Prototype

The electronics used to conduct lab testing of the miniature replica turbines was reconfigured and mounted in a weather-proof pelican case (Figure 31). The addition of a GPS module was made to synchronize the time clock to the global positioning satellites. This universal time stamp was later used to precisely correlate torque and wind velocity data to the data collected from the weather station located at the URI Bay Campus.

The portable enclosure was also outfitted with a micro SD card shield and 10,000 mAh battery for data storage, away from a monitoring PC. Along with the GPS module, a WattsOn power transducer was added for future power testing of the turbine setup. The power transducer was coupled with a MODBUS Arduino shield for communication of power transmission data. The power mon-

itoring hardware was not used in the scope of the experiments described in this literature.

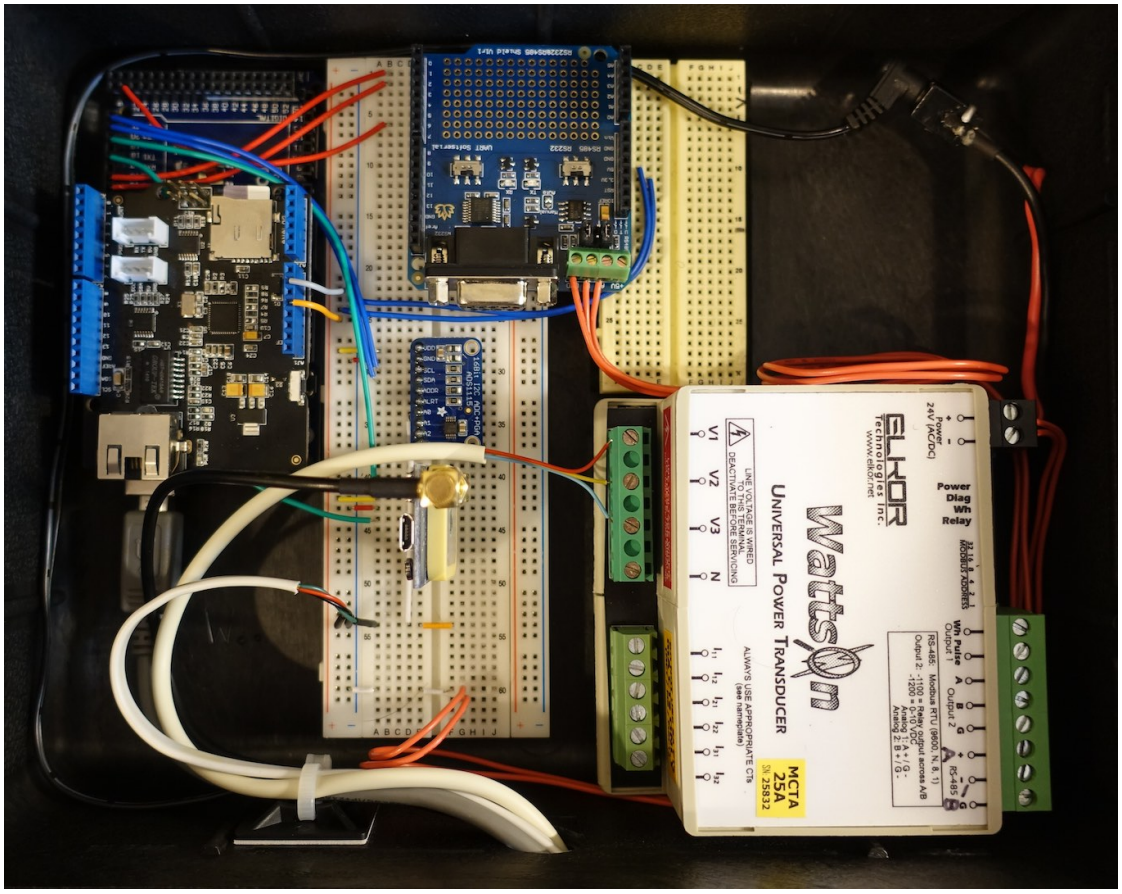


Figure 31: Data acquisition hardware mounted in a weather proof Pelican enclosure

5.3 Software

An Arduino control system routine was laid out in a block diagram (Figure 32) to effectively capture select turbine data at precise intervals and transmit the data serially to a Visual Basic server program running on a PC. The routine utilizes an interrupt running with the Arduino clock. At each interrupt, specific conditions are checked. If the Arduino has received a run command from the user, the capture rate setting is compared to the current time. If this time has been exceeded, pressure and torque data will be gathered and sent to the server. The

full program can be found in Appendix A.

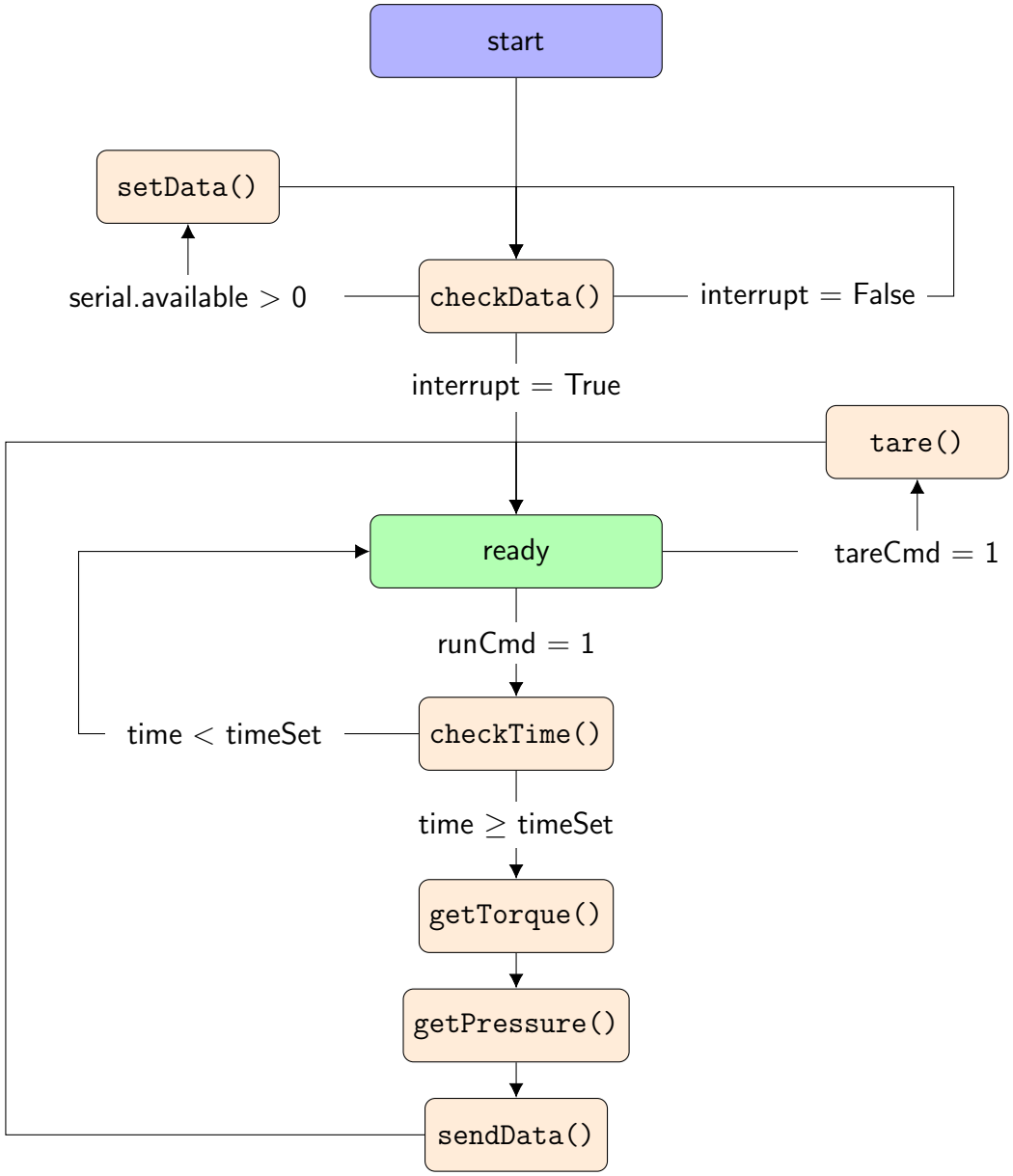


Figure 32: Arduino control system block diagram for static torque and wind velocity capture

The server PC is running a data acquisition program written specifically to interface with the Arduino and capture data from the selected components (Figure 33). The user is able to select a COM port for the Arduino serial communication

and set a data capture interval. The location and file name, of the comma separated data file, can also be set. The user is able to start and stop data feedback and tare the load cell before capturing data to a file. The time can then be zeroed and capture will begin until stopped by the user. The full Visual Basic program can be found in Appendix B.

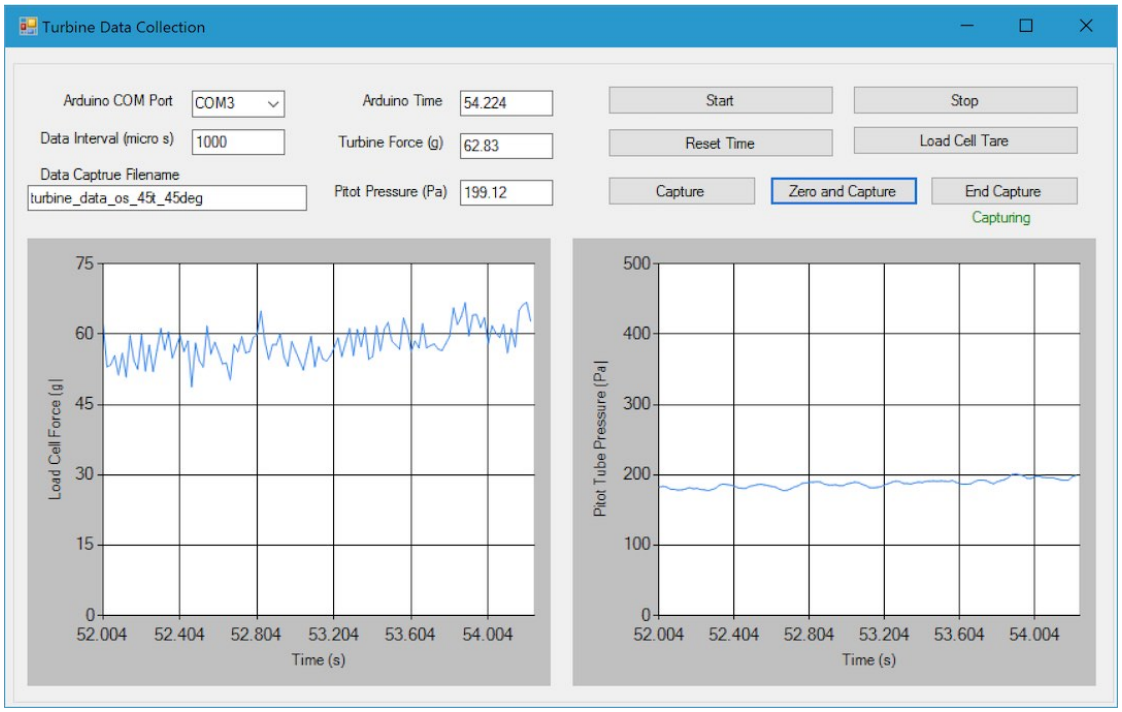


Figure 33: Visual Basic graphic user interface for Arduino data acquisition

CHAPTER 6

Testing

6.1 Calibration Procedures

Before testing could commence, calibration was carried out on all load cells and wind measuring devices. To calibrate the load cells, precision weights were used. First, an unloaded reference voltage was recorded from the ADC. Then a 10-gram weight was placed about the center of the load cell pinning point and another voltage reading was taken (Figure 34). Next, a second 10-gram weight was applied and the values were plotted to give a voltage versus load curve. The linear relationship allowed for the slope to be taken and used as a calibration factor for converting load cell voltage to a gram measurement.

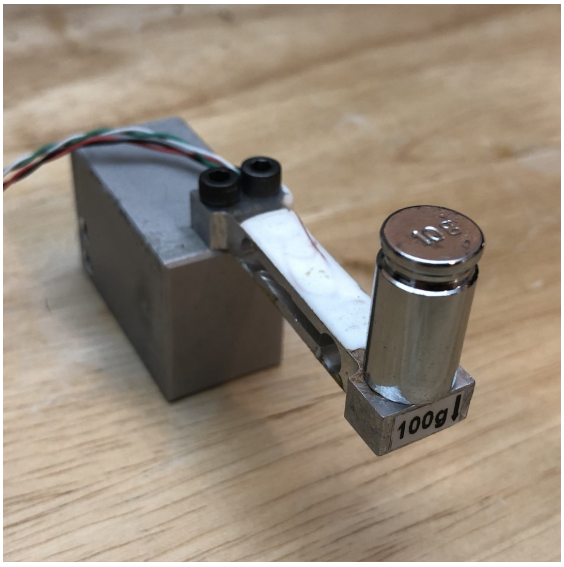


Figure 34: 10-gram calibration weight centered over pinning point of a 100-gram load cell

Following the calibration procedure for the load cell, a similar process was used to calibrate and convert the differential pressure sensor readings into velocity measurements. An Extech manometer was used to precisely set the wind velocity

in the Aerolabs wind tunnel. The static pitot tube was then placed into the stream with flow directed parallel to the inlet. Multiple voltage readings were taken from the pressure sensor, each corresponding with a recorded wind velocity. The slope of the velocity versus voltage curve was then used to define a calibration factor for the differential pressure sensor with static pitot tube attached. This procedure was repeated for the Adafruit anemometer.

The digital step count was then converted to a velocity through the following procedure. Given that the 16 bit analog to digital converter has $2^{16} = 65,536$ discrete steps from -2.048 to 2.048 volts, that gives 6.25e-5 volts/division. With the anemometer output ranging from 0.4 to 2 volts over 0 to 32.4 m/s, we have a coded range of 6,400 = 0 m/s to 32,768 = 32.4 m/s. This gives a conversion of Equation 2.

$$(N - V_{RefLow})(\frac{v_{max}}{E_{FSR}})(\frac{V_{RefHi}}{2^M/2}) = v \tag{2}$$

where N is the number of discrete measurements, M is the number of ADC bits, E_{FSR} is the full scale voltage range, v_{max} is the greatest measurable wind velocity, v is the measured wind velocity, and V_{RefLow} and $V_{RefHigh}$ are the minimum and maximum voltages which can be coded, respectively. E_{FSR} is calculated in Equation 3.

$$E_{FSR} = V_{RefHi} - V_{RefLow} \tag{3}$$

Therefore, the wind velocity measured by a coded ADC output from the anemometer is given by Equation 4.

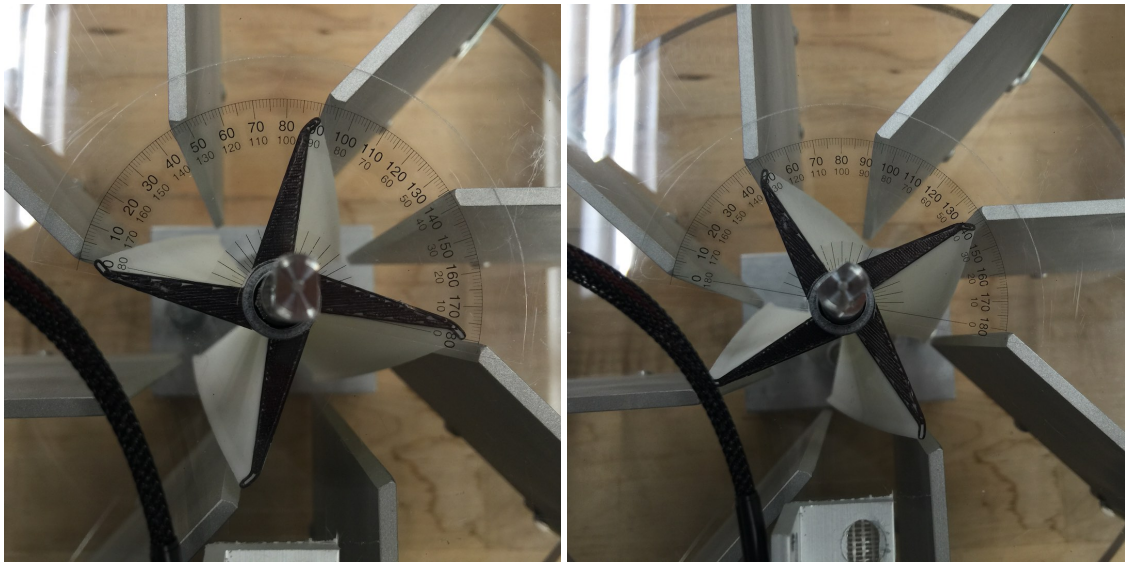
$$(N - 0.4)(\frac{32.4}{1.6})(\frac{2.048}{2^{16}/2}) = 1.266\text{e-}3N - 5.06\text{e-}4 = v \tag{4}$$

6.2 Static Testing

6.2.1 5-inch Turbine Replica

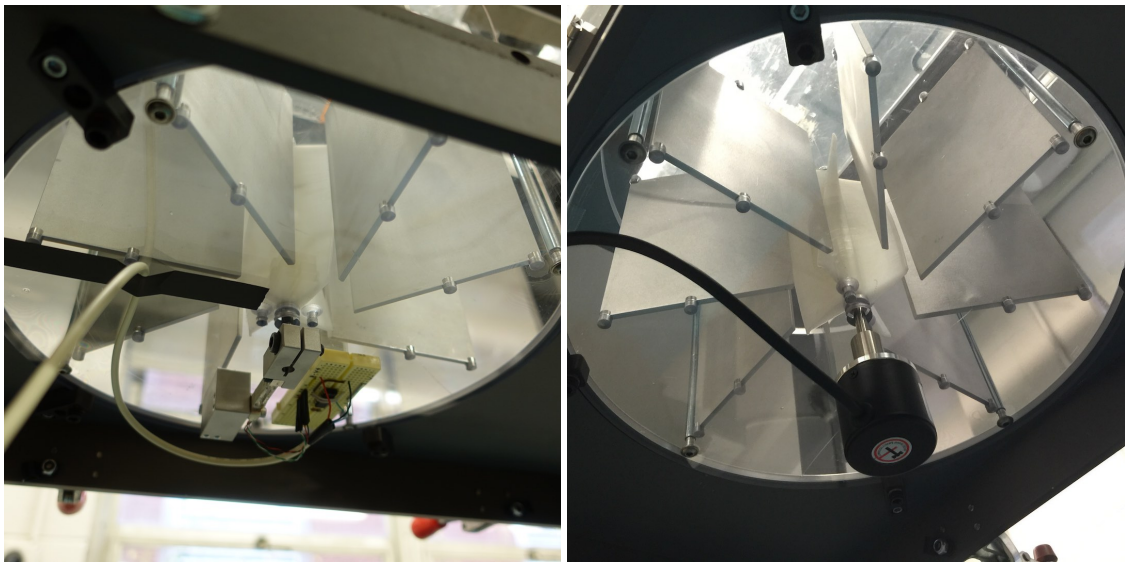
Testing began with a static torque analysis. This was achieved by incrementally setting the turbine position a number of degrees from the lower turbine tip

to the windward stator edge. Figure 35a shows a setting of 0° offset and Figure 35b shows a 50° offset. These settings were locked in using the clamping load cell mount (Figure 36a). The initial test procedure was to ramp the wind tunnel velocity to 20 m/s over a 60 second duration. The turbine was positioned so that the angle measuring reference-stator was parallel with the flow direction of the wind tunnel. During the ramp time, torque and wind velocity data were recorded 50 times per second. This was repeated for all stator and turbine combination at 10° stator-turbine offset increments from 0° to 80°, producing 54 data sets.



(a) (b)

Figure 35: Static torque turbine offset settings from windward stator. (a) 0° offset and (b) 50° offset



(a) (b)

Figure 36: Wind tunnel turbine setup for (a) static torque and (b) angular displacement

With every combination of stator and turbine captured, the worst performing offsets of the cases which could be most easily recreated by the 2.5-inch replica and 7-foot prototype, were selected for more accurate analysis. A second test procedure was developed to more exactly capture the torque versus wind velocity curve. This procedure consisted of holding the turbine stationary, at the lowest performing torque case, while immersing the turbine in a constant flow field (Figure 37). Data was then recorded for 60 seconds at 50Hz, producing a cloud of points representing the torque for a given velocity. The procedure was the repeated for 20 increments from 0 to 20 m/s, wind speed.

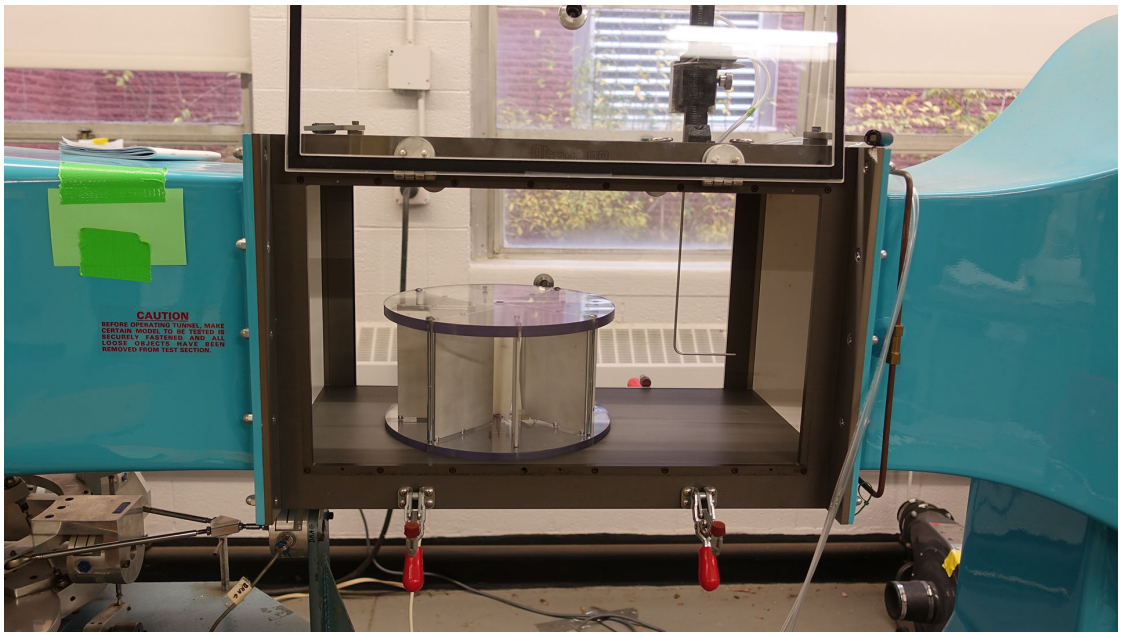


Figure 37: 5-inch replica in the Aerolabs wind tunnel set for static torque measurements

6.2.2 2.5-inch Turbine Replica

After subsequent testing to determine the output for combinations of stators and turbine helices for the 5-inch model, a test with straight stators and 45° turbine was conducted for the miniture, 2.5-inch replica. Given the lower torque output, the model was tested statically to a wind velocity of 35 m/s (Figure 38). The turbine was tested at the worst performing offset angle for the 5-inch model. This data is later used in scaling equations to compare velocity and torque outputs for the given sizes. These results were then each be scaled for comparison to the 7-foot protoype.

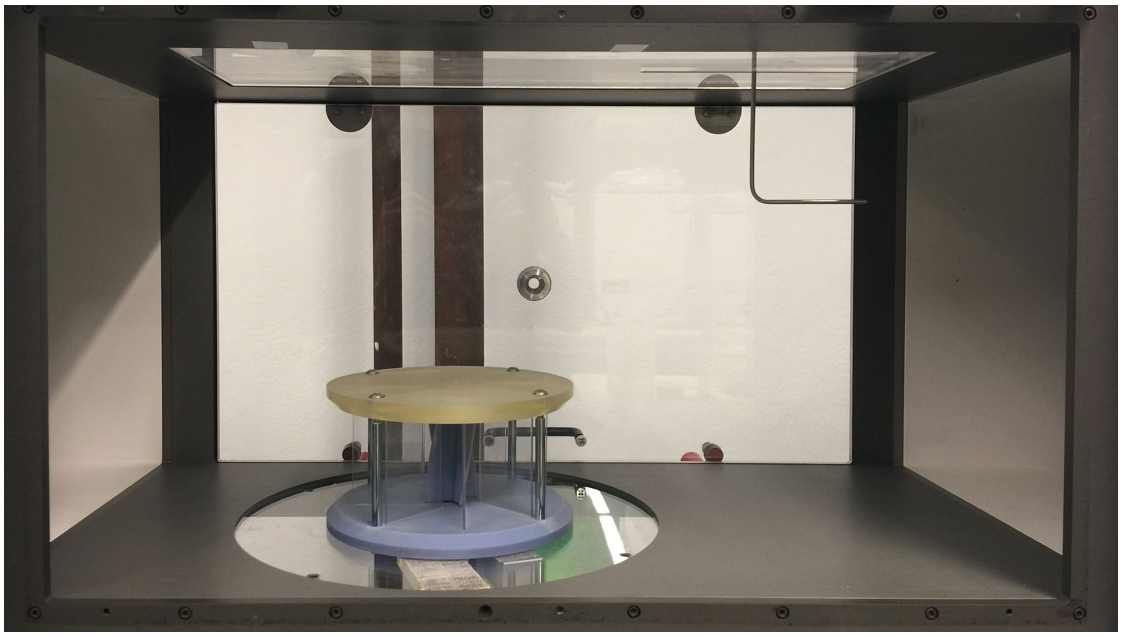


Figure 38: 2.5-inch replica in the Aerolabs wind tunnel set for static torque measurements

6.2.3 7-foot Turbine Prototype

Despite the limitations in wind control for an outdoor test setting, comparable torque versus velocity data needed to be acquired for the full-size prototype. This gives way to the logic behind examining the models in their worst performing cases. With the worst model cases used for prediction, output from the 7-foot prototype will always exceed the scaled results from the two replicas, no matter the wind direction. The equivalent test was carried out on the 7-foot prototype by mounting the 50 kg load-cell to the inner base of the turbine housing. The distance was fixed from the axis of the turbine at 18-inches or 0.4572 m and contact was made to the back of a turbine blade. The turbine was then preloaded with elastic cord to eliminate bounce. The preload was zeroed out of the force measurement and data points were recorded continuously at 4 Hz for 12-hour sessions. The

torque calculation is expressed by Equation 5.

$$\tau = N \times c_{sf} \times L_{ma} \tag{5}$$

where N is the number of steps output by the ADC, in response to the load cell, c_{sf} is the scale factor calculated during calibration and L_{ma} is the length of the moment arm contacting the load cell.

For supporting wind velocity data, collection was conducted from the URI weather station, which was located nearly 30 feet from the ground. This was accompanied by a WindLOG assessment tool set atop a 5-foot pole in an open area. Finally, the stand-alone anemometer was mounted in line with the center of the turbine height, placed on the flat region of the carrying trailer. All devices were within a 50-foot radius of one another and differed primarily by mounting technique (Figure 39).

Figure 39: Testing location map of the URI Bay Campus, 7-foot turbine layout

6.3 Dynamic Testing

Dynamic lab testing was performed only on the 5-inch replica to achieve a performance benchmark used in turbine and stator combination selection. With optimized stator and turbine helix combinations found for the 5-inch model, Similarity Theory suggests it can be expected that the behavior will be equivalent for alternately sized units (Foken, 2006).

6.3.1 5-inch Turbine Replica

An analysis of the static torque readings could not define an optimal configuration alone. The angular velocity at a given wind speed needed to be compared between combinations, along with the angular acceleration from release, in a steady flow condition. For the angular velocity case, the wind tunnel was ramped to 20 m/s over 60 seconds. During this time, the velocity of the turbine was measured using the attached encoder. This was repeated four times for each of the six cases.

The acceleration test was performed by placing the turbine replica in the wind tunnel test chamber and setting the wind speed to a nominal 10 m/s. The output shaft was held fixed until a stabile velocity was reached. At this point, data recording commenced and the turbine was released. Angular displacement was then measured every 0.02 seconds until a steady state was achieved.

6.3.2 7-foot Turbine Prototype

The dynamic tests performed on the 7-foot prototype was established to prove the efficacy of the model scaling equations in predicting a 5 kW generator, starting torque and subsequent wind speed. The approach involved monitoring the angular displacement of the turbine for any change during a fraction of a second. If a change was detected, data recording began for a minimum of 30 seconds, but would continue until motion stopped if continuous.

CHAPTER 7

Results and Discussion

7.1 Static Test Results - 5-inch Turbine Replica

The raw data plots for all static torque readings and their corresponding configurations can be found in Appendix C. For the purpose of selecting the optimal stator and turbine helix, all turbine offsets were averaged and the data smoothed using a gaussian moving average with a window length of 50 data points, or 0.2 m/s. From the results in Figure 40, it is clear that the straight turbine blades had the worst overall performance. The addition of the helix brought performance to a nearly comparable level for all remaining cases. This data confirmed the leading combination to be the open stators with 45° offset turbine, but the margin appears small.

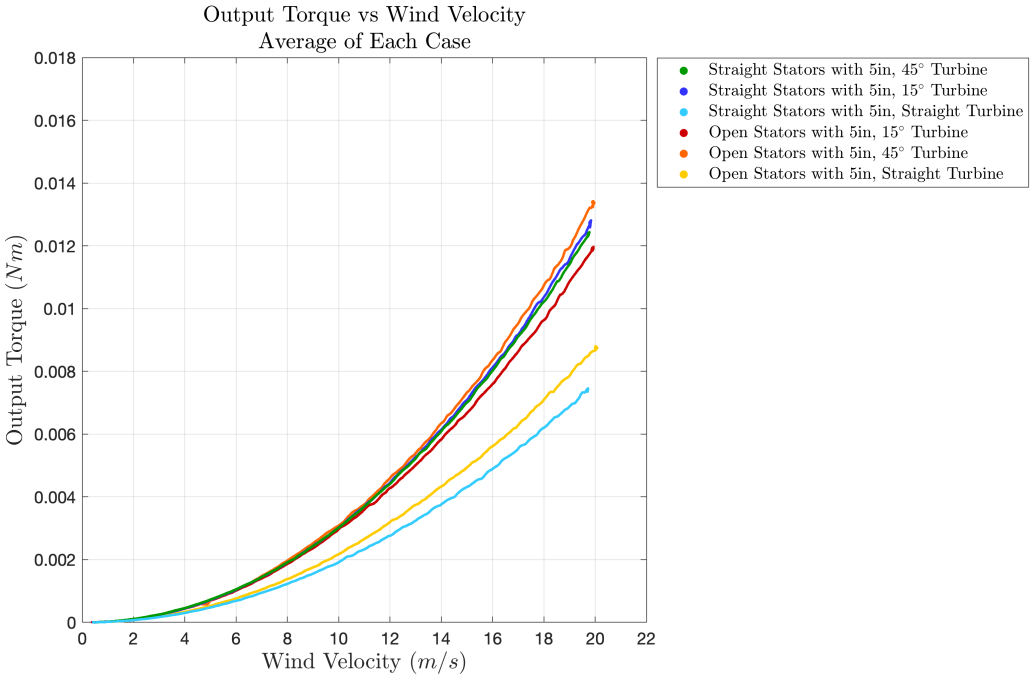


Figure 40: Output torque vs wind velocity of values averaged for all offsets of each turbine and stator configuration

However, when comparing the worst performing offset angels, from each case (Figure 41), the separation in performance is greater. In this plot of lowest torque offsets, a more clear distinction of the benefits is realized for the combination of open stators and 45° turbine. The importance of considering this case lies in the occurrence of stalling conditions when the turbine comes to rest at such angles.

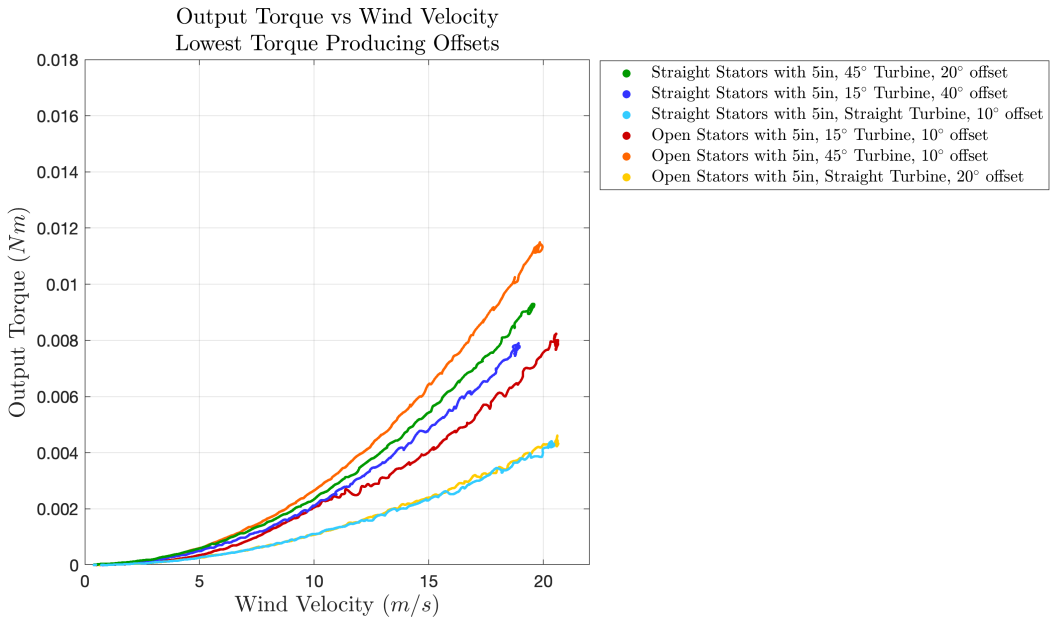


Figure 41: Output torque vs wind velocity of the worst performing offset angles for each turbine and stator configuration

The condition is further exemplified when zooming in on the low wind velocity regions of the plot (Figure 42). This portion of the curve represents the effective wind velocities which will start the rotation of the turbine. We see here that the greatest contributing factor to starting torque is the addition of the helix angle to the turbine. When considering the total positive influence of design parameters, the addition of open stators does increase static torque production at higher wind velocities (Figure 41). However, the functional benefit of this increased torque must be further analyzed in the dynamic testing cases.

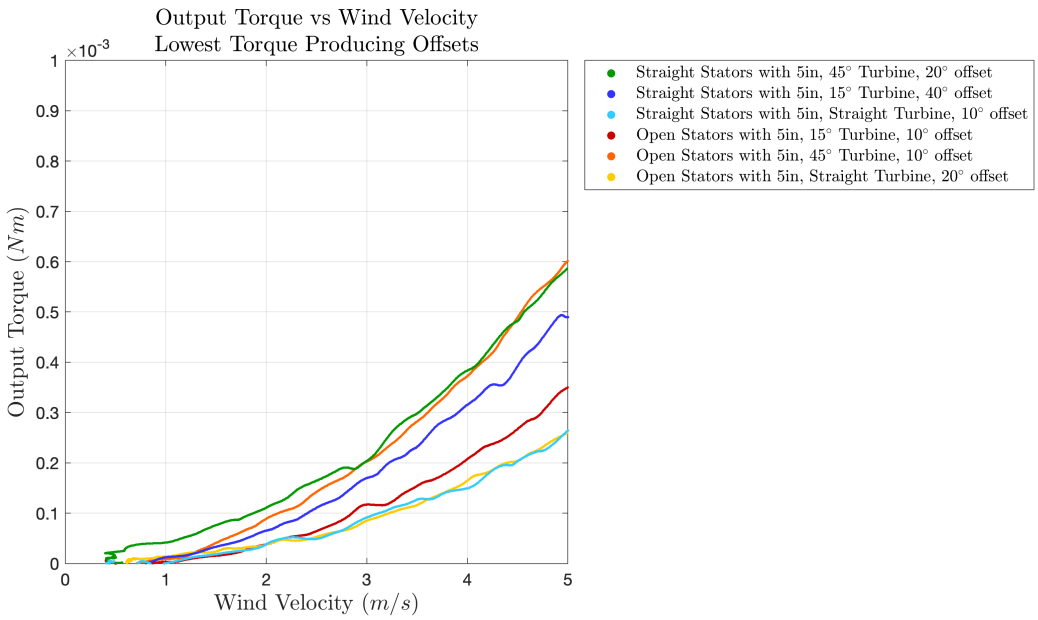


Figure 42: Starting region of output torque vs wind velocity of the worst performing offset angles for each turbine and stator configuration

Before making a definitive conclusion of the most viable turbine configuration from the static torque data, another result must be considered. While averaging the torque and wind velocities makes for a readable plot, performance conditions may be lost. Figure 43 shows the three top performing combinations and their averages between offsets, with the inclusion of standard deviation over averaged offsets. This demonstrates a more consistent output torque from the open stator and 45° turbine combination, than the next closest contender of straight stators with 45° turbine. It can be speculated that a more consistent torque output with relation to angular position would result in a greater rotational force with lower interference.

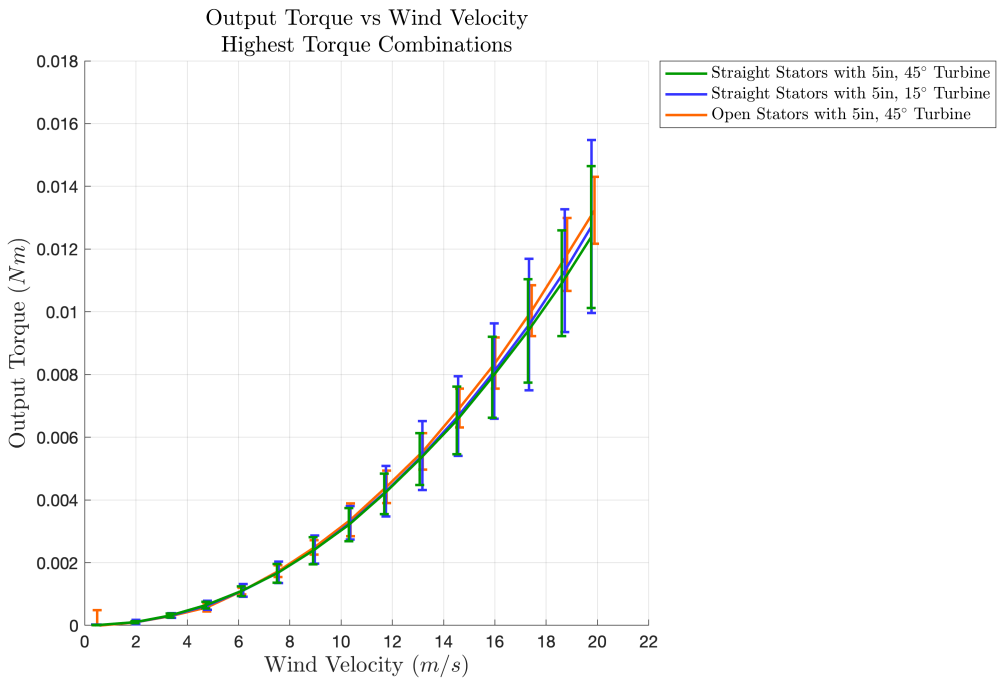
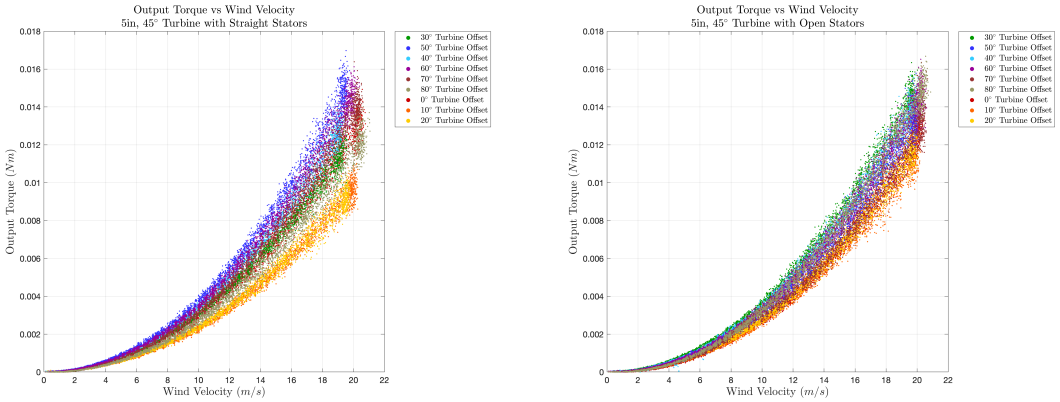


Figure 43: Output torque vs wind velocity of the top performing turbine and stator configurations with standard deviation

This rational can be further exemplified by reviewing the raw data for the two 45° turbine cases (Figure 44). The wider variation of torque output values from the straight stator case would indicate less consistency in rotational forces, suggesting the open stator design will produce superior dynamic performance.



(a) (b)

Figure 44: (a) Straight stators with 45° turbine and (b) open stators with 45° turbine

Although local pressures were not measured in the scope of these experiments, Figure 45 exemplifies what may be the result of added surface eddies. Each of the six data sets is for a turbine-stator case, all at a single 20° offset. The greater variation in output torque, for a given wind velocity, of the straight stator cases, indicates a more unsteady flow state. The cause of this variation is a subject for future research, but for the purposes of this analysis, the more stable response is deemed a positive feature.

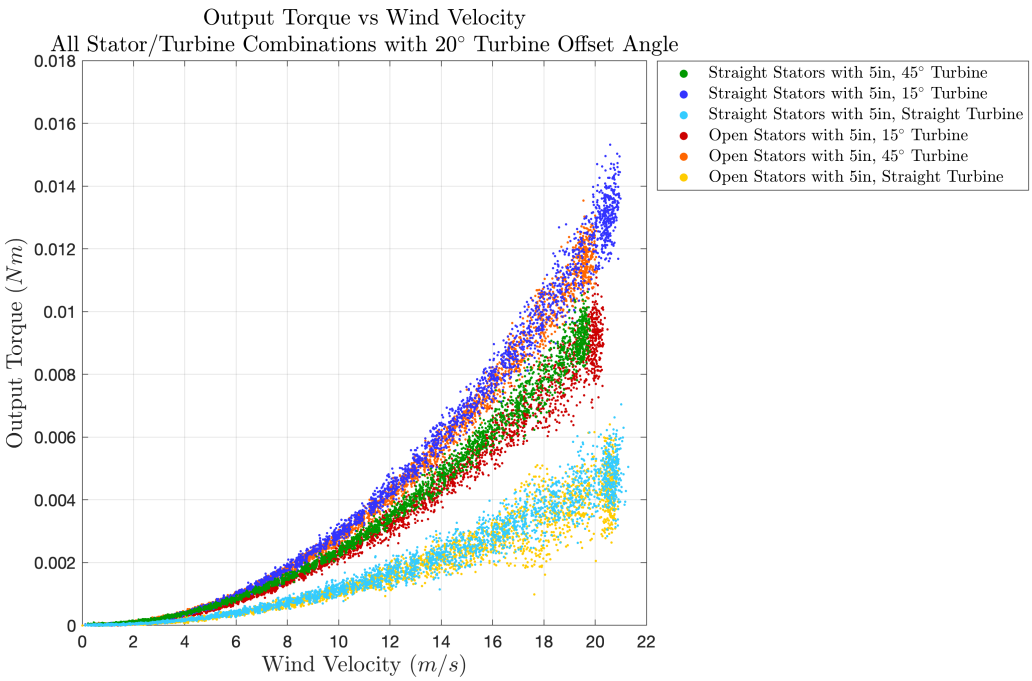


Figure 45: Output torque vs wind velocity for all turbine and stator configurations with at a 45°

Finally, the results for the improved accuracy test of the lowest torque producing offset, for the straight stator and 45° turbine case can be seen in Figure 46. This case was chosen for further scrutiny and scaling due to it's high performance and ease of replication in the 2.5-inch model and 7-foot prototype. With over 1,500 data points captured in a 60 second period, for each of 20 wind velocities, more than 30,000 total points can be seen plotted. The fluctuation in measured output torque for a given velocity is more clearly represented with each data cloud.

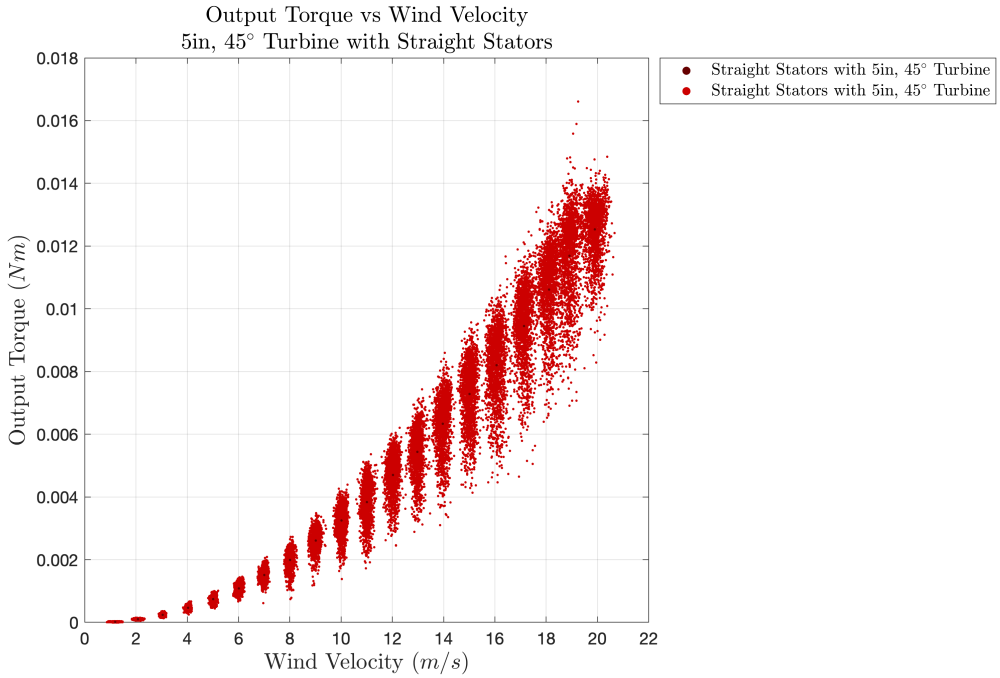


Figure 46: Output torque vs wind velocity for low performance offset angle from large data sets

The average of each cluster was calculated and connected in Figure 47, with the inclusion of the standard deviation. This curve represents the most accurate performance measure of the static torque output from the straight stator and 45° turbine case, which will be studied further in the 2.5-inch and 7-foot turbines.

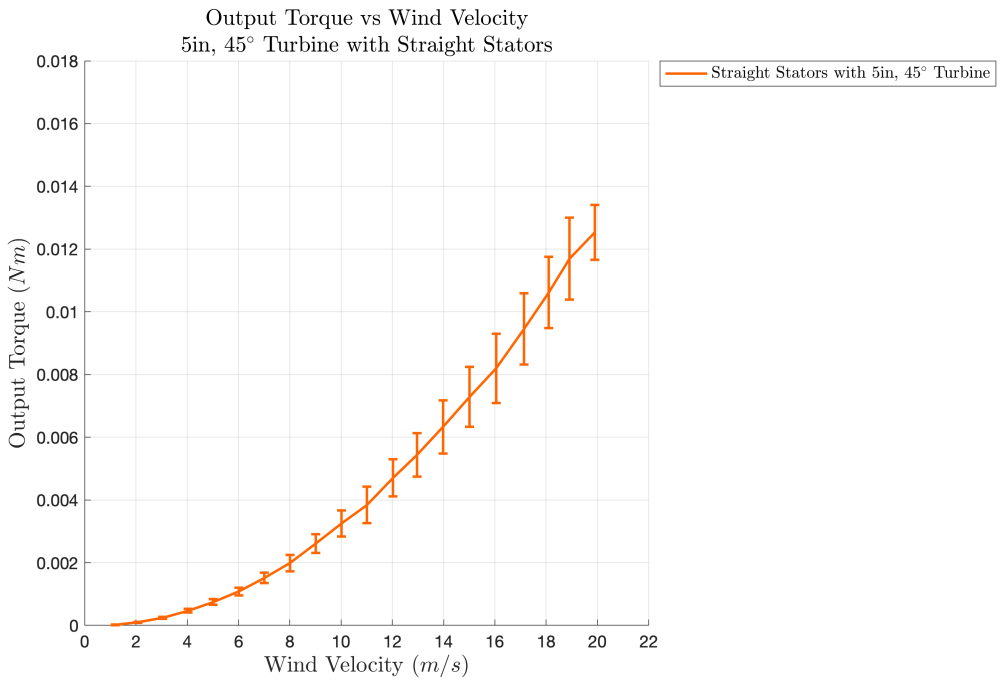


Figure 47: Output torque vs wind velocity for low performance offset angle from large data set averages with standard deviation

7.2 Dynamic Test Results - 5-inch Turbine Replica

To further understand how increased static torque translates to dynamic performance, we first review angular velocity versus wind velocity as seen in Figure 48. This plot demonstrates that the straight turbine cases require a greater wind velocity to achieve the same rotational speed. More importantly, the spotty vertical connecting lines represent start points where the turbine was stuck in a stalled state. Given the four overlayed data sets for each case, we can see the propensity for stalling for each.

While the open stators and straight turbine slightly outperform the stalled 45° turbine, it only does so marginally. Considering the predominately lower starting torque of the 45° turbine cases, along with greater achieved velocity for a given wind speed, the 45° offset helix is deemed the ideal turbine design, of those tested.

The considerably close performance between opened and closed stators brings to question if the marginal gain of the design exceeds the marginal increase in cost and complexity.

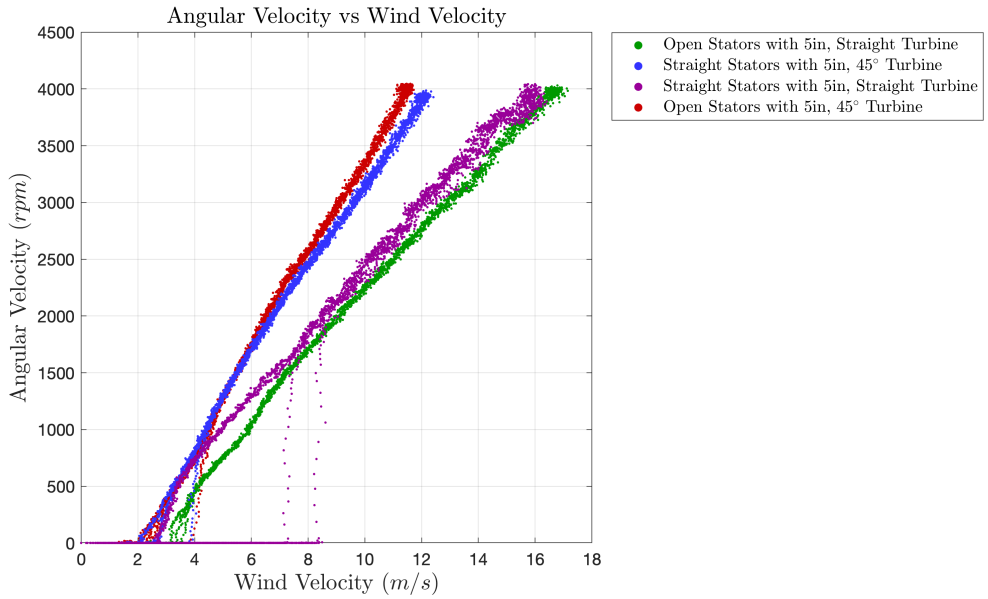


Figure 48: Angular velocity vs wind velocity for variable velocity

The second dynamic test of angular velocity versus time, for a constant wind speed, is show in Figure 49. For these results, the wind tunnel was set at a constant wind speed and the turbine was released from rest. This plot accentuates the increase in rotational velocity for the same wind speed, showing an average improvement of 11% or 300 rpm. Despite the greater maximum velocity, the straight stators and 45° turbine achieved a steeper slope, indicating greater acceleration.

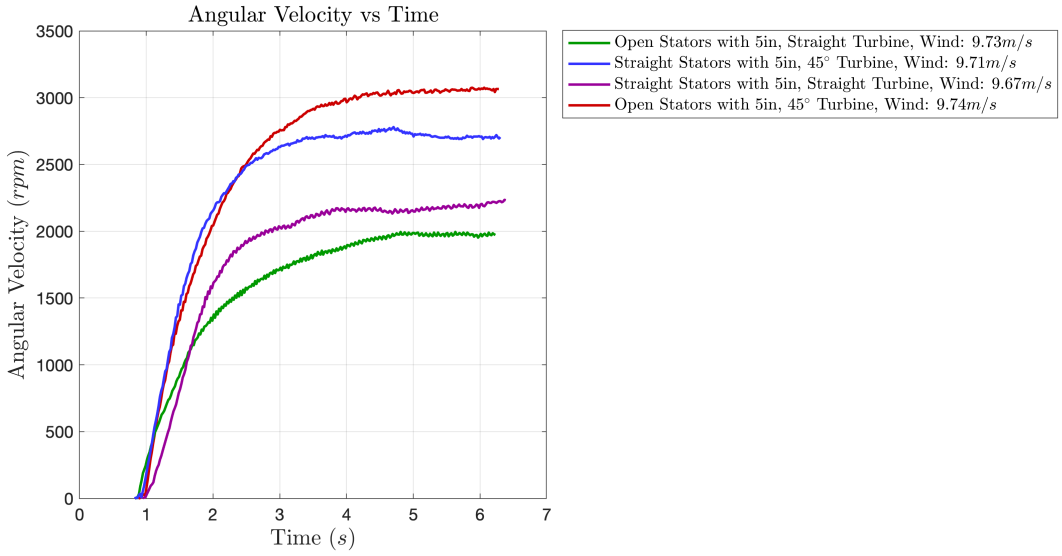


Figure 49: Angular velocity vs time for constant wind velocity

The angular acceleration versus time was plotted in Figure 50. These curves show the slightly greater acceleration achieved by the straight stators and 45° turbine. The trade off is clear that while greater acceleration is achievable with straight stators, this comes with a greater wind speed needed to achieve the same velocity.

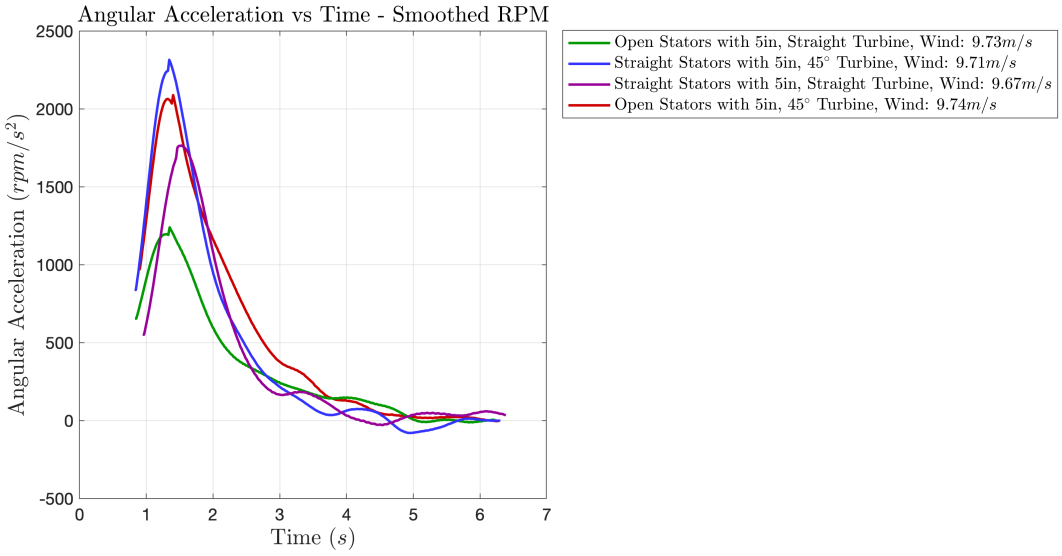


Figure 50: Angular acceleration vs time for constant wind velocity

7.3 Static Test Results - 2.5-inch Turbine Replica

With an understanding of which turbine configuration would perform optimally in low wind conditions, the selected configuration of straight stators with a 45° offset was pursued for the 2.5-inch replica testing. The purpose of the miniature replica was only for the validation of torque scaling equations, used to predict starting torque. Therefore, only a single static torque case, corresponding to the high density data test done on the 5-inch, straight stator and 45° turbine, was performed. The results for both the 2.5 and 5-inch turbine can be seen in Figure 51 and 52.

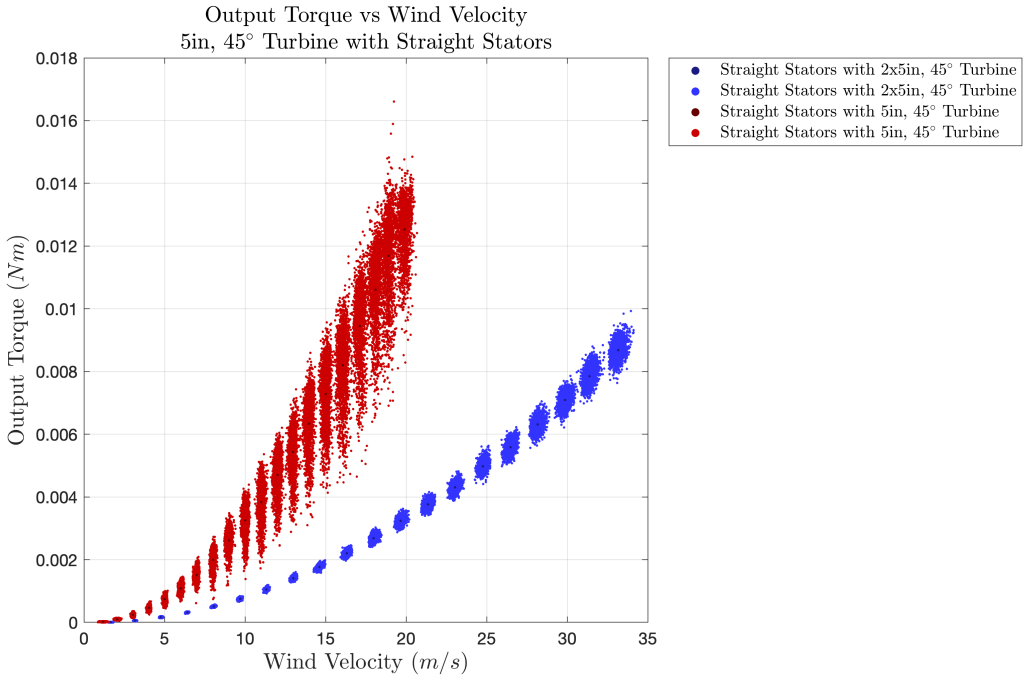


Figure 51: Output torque versus wind velocity of 2.5 and 5-inch replicas with straight stators and 45° turbine

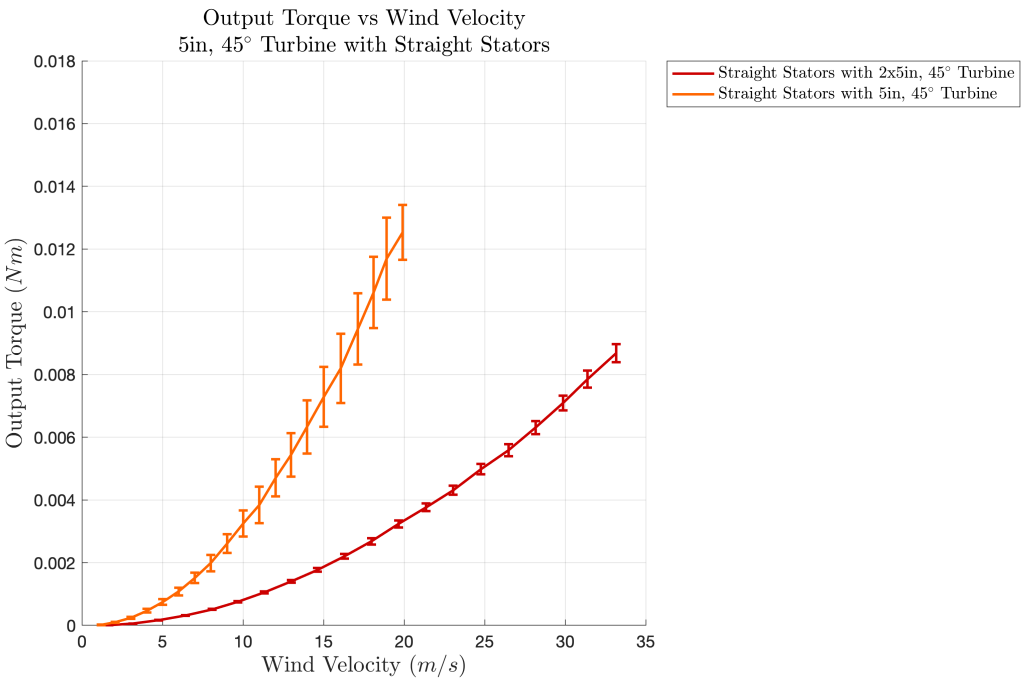


Figure 52: Averaged output torque versus wind velocity of 2.5 and 5-inch replicas with straight stators and 45° turbine, with standard deviation

7.4 Static Test Results - 7-foot Turbine Prototype

As discussed, the test setup for the 7-foot turbine was less controlled due to the real world environment it was placed in. The data proves to be more crude, but contains the plot cloud of torque output versus wind speed, for comparison to the 2.5 and 5-inch replicas. Of primary consideration are the lowest torque values, which nearly form an exponential curve similar to the wind tunnel data.

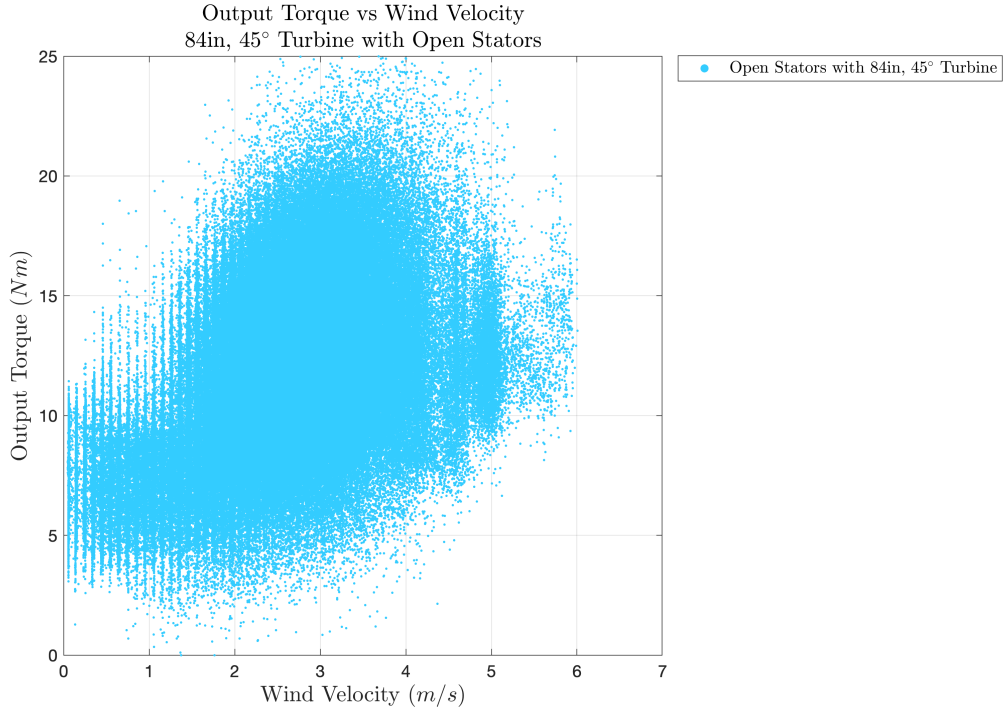
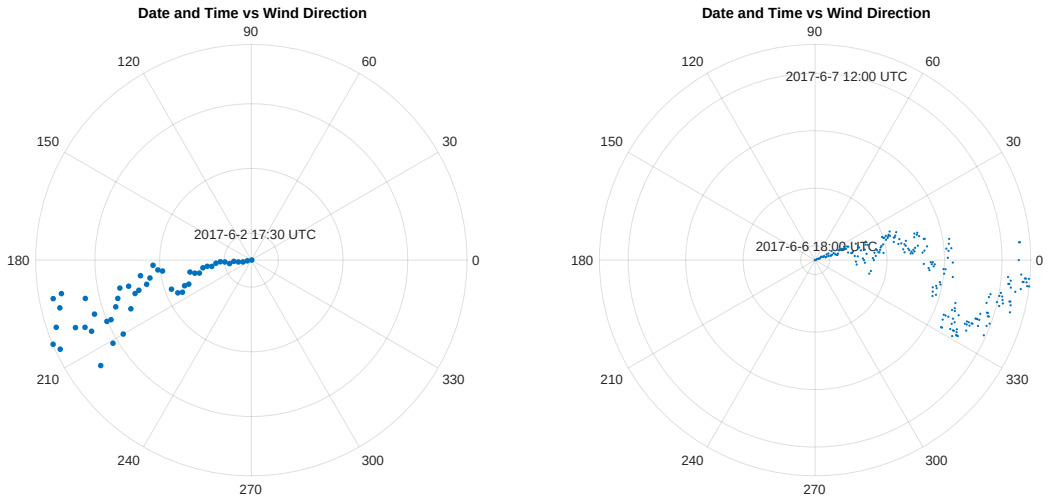


Figure 53: Output torque versus wind velocity of 7-foot prototype with straight stators and 45° turbine

When analyzing the torque versus displacement plot, we see torque values logged for nearly zero wind velocity. This occurrence is due to the response time of both the load sensor and anemometer. While the rotation of the anemometer must overcome the inertial resting force of the rotor, the load cell is already in contact with the turbine blade, with a preload. This results in an immediate rise in torque reading, while the anemometer registers an increase in wind velocity just slightly later. Therefore, we see a rise in torque before the anemometer has a chance to respond. This suggests that constant wind velocities will provide more accurate readings over gusts. Additionally, the anemometer requires a 0.2 m/s wind velocity to start, while the load cell registers for every wind speed.

7.5 Dynamic Test Results - 7-foot Turbine Prototype Coupled to Generator

Two primary days of dynamic testing took place on June 2nd and 6th of 2017. The wind direction was logged and mapped using the weather station at the URI Bay Campus, over the duration of the 12-hour testing (Figure 54). The turbine prototype was set with a stator aligning with the headwind every 45°, beginning with due North. With the wind fluctuating within a $30-60°$ arc, some consistency existed despite the outdoor arrangement.



(a) (b)

Figure 54: Plolar plots of wind direction during dynamic 7-foot turbine testing

A comparison was conducted on the variations of measured wind velocities from the three sources. This data shows the discrepancies produced by changes is source height and surroundings (Figure 55). While the anemometer had the lowest readings for the same wind state, this value was representative of the condition seen by the turbine, with minor impedance of surrounding obstacles.

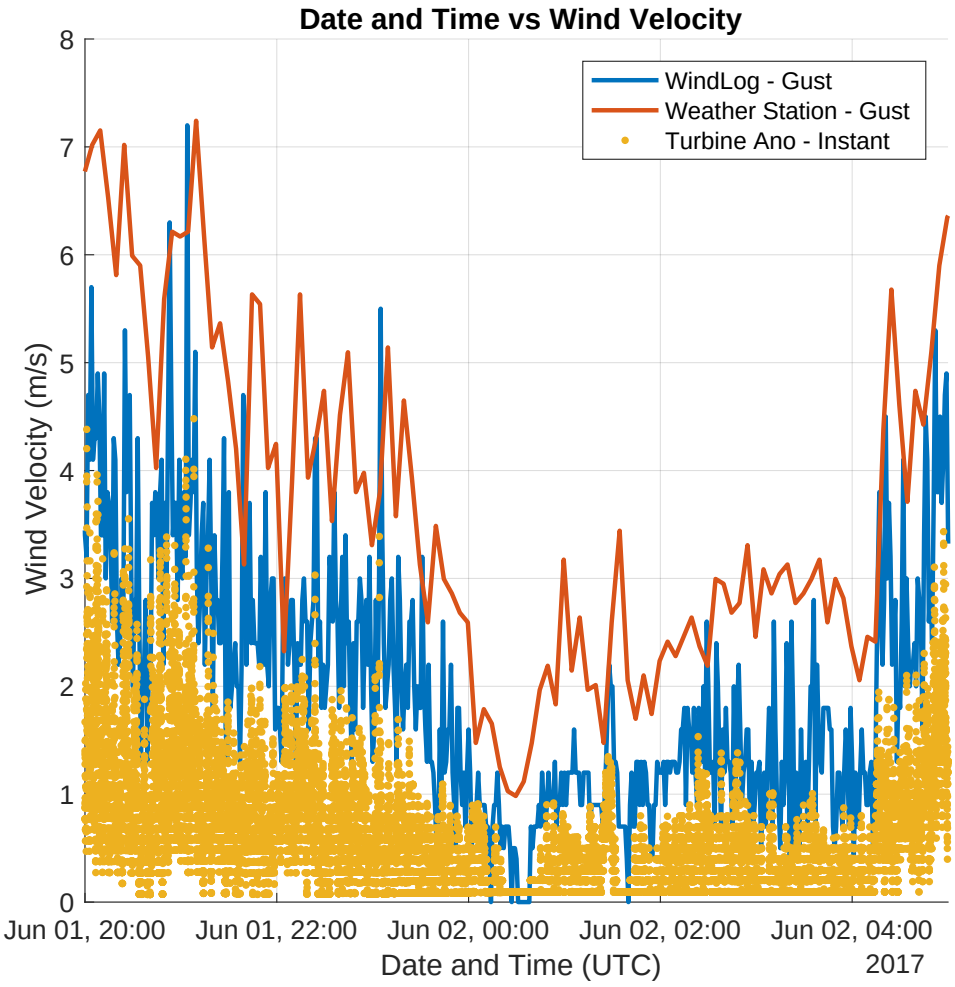


Figure 55: Wind velocity variations between three sources, URI Weather Station, WindLOG and anemometer

A consistent segment of wind direction was selected to evaluate the turbine starting wind speeds, over a period of time which contained many start and stop cases. The turbine was coupled with a 5 kW generator, with no electrical load. Figure 56 shows a 3-hour snippet of time where the wind speed was captured whenever the turbine began rotating and continued for two seconds. The results show an average starting speed of 1.99 m/s, which was consistent with visual

observation. A number of difficult start combinations led to a few outliers reaching the 6 m/s mark.

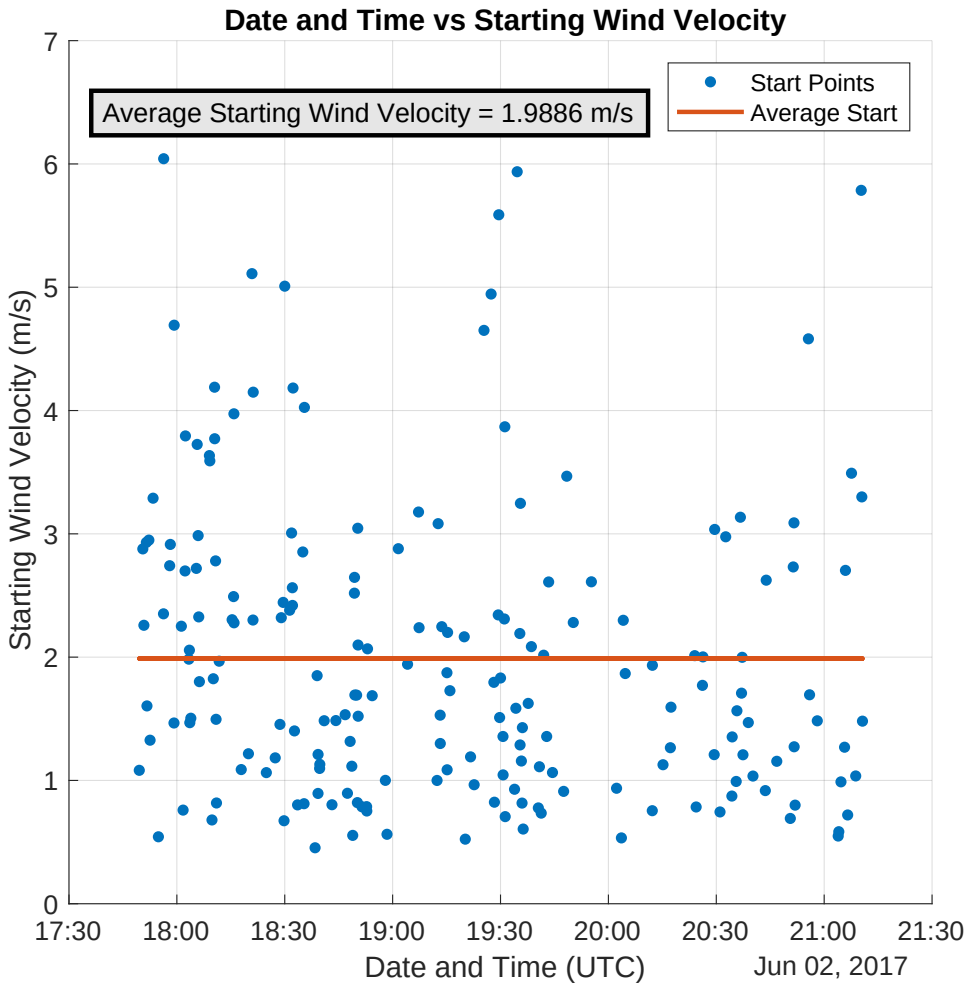


Figure 56: Turbine starting wind speed versus time for a 3-hour segment of frequent start and stop cases

A single start-stop case was plotted in Figure 57. This shows the relative turbine velocity for a given wind speed over a small period of operation. The turbine velocity can be seen responding at a delayed rate to fluctuations in wind velocity. This is to be expected given the massive increase in inertia from the

anemometer rotor to the 7-foot turbine.

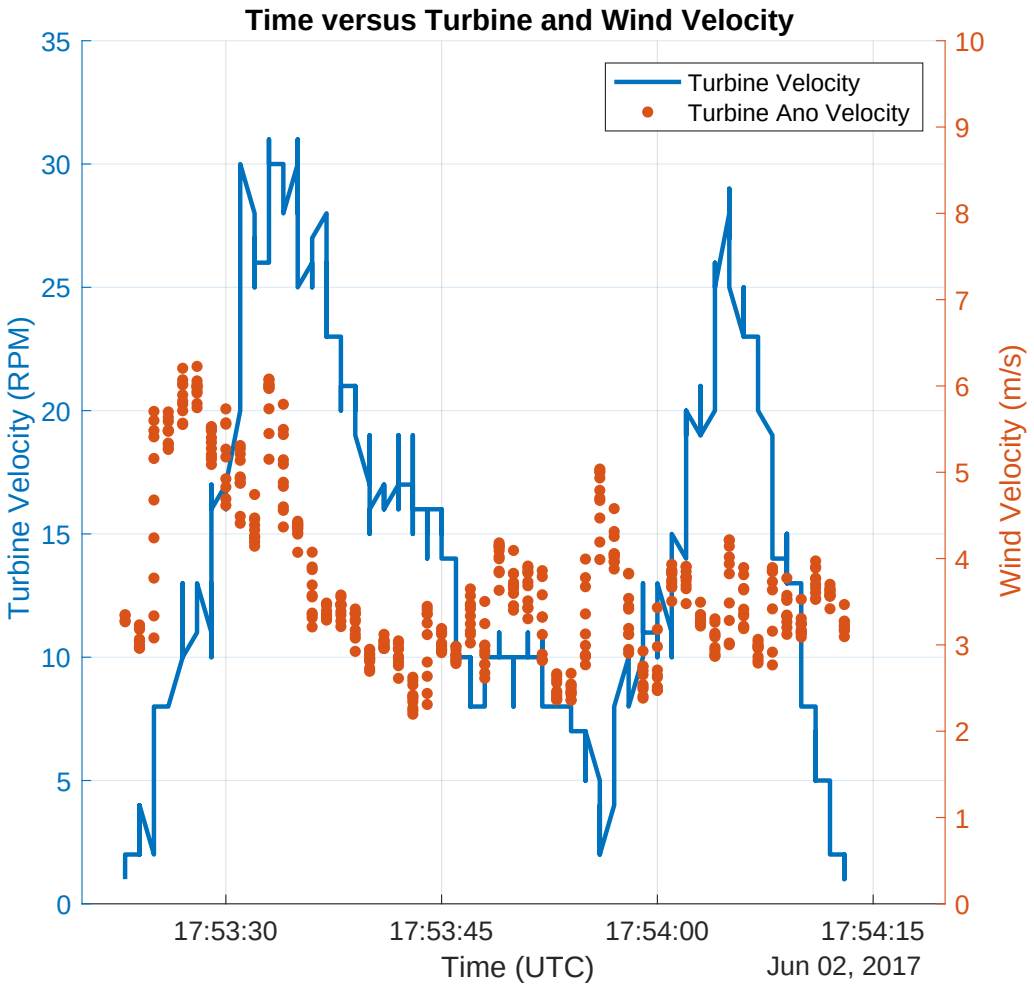


Figure 57: Turbine angular velocity and wind velocity versus time for a single start-stop case

During the longest recorded period of continuous rotation (Figure 58), lasting a total of 28-minutes, a trend was also evident, but less descriptive. However, even with the condensed data, the correlation of turbine velocity and wind velocity can be observed. This data exemplifies the potential angular velocity achievable by the 7-foot prototype. This case is only turning the generator with no electrical load,

so results are expected to be far less with any resistance from the generator.

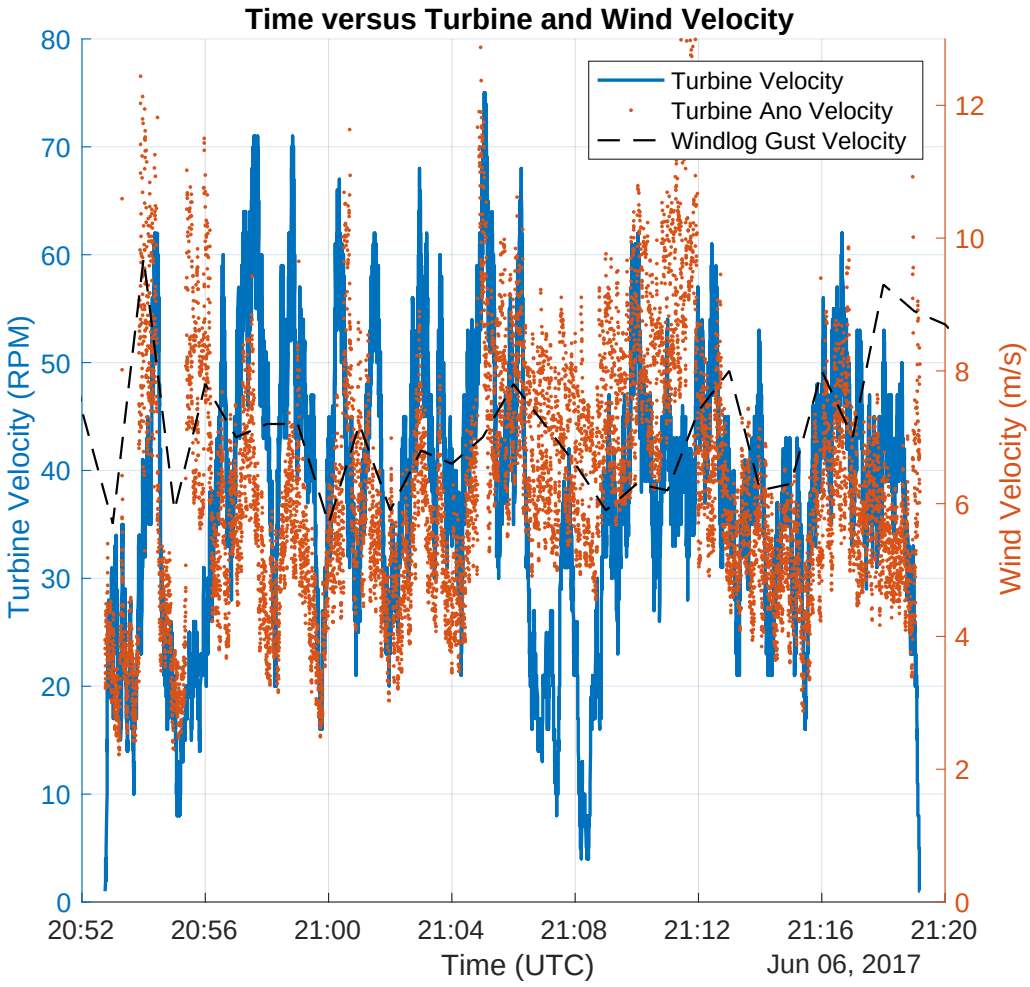


Figure 58: Turbine angular velocity and wind velocity versus time for a single start-stop case of maximum duration

Finally, all the rotational data can be seen plotted a together versus wind speed in Figure 59. This plot shows the starting span ranging from 1 to 5 m/s with unloaded angular velocities peeking at 100 rpm. While the significance of achieving high rotational speeds of an unloaded generator is questionable, starting the generator with an average wind speed of 1.99 m/s is significant, given the

predictions of starting torque computed in the following chapter.

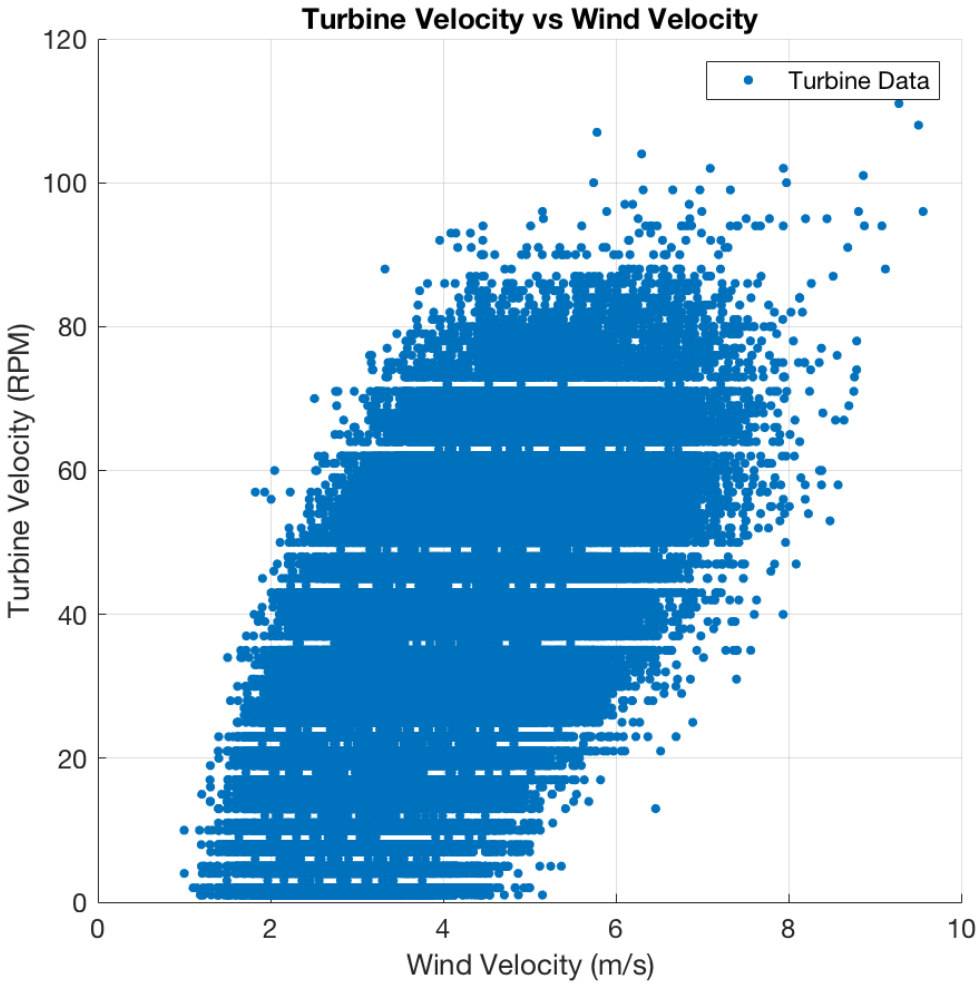


Figure 59: Turbine angular velocity versus wind velocity for all data points

CHAPTER 8

Scaling Predictions

One of the primary objectives this research aimed to reach, was the ability to predict the wind speed required to start a 5 kW generator, with no load. The effective breaking torque of the generator, without an electrical load, was specced to be about 1 Nm. A scaling comparison between full size and model geometry was derived by way of non-dimensional analysis using the Buckingham Pi Theorem. The results can be seen in Equations 6 and 7.

$$U_f = (\frac{\mu_f}{\mu_m})(\frac{L_m}{L_f})(\frac{\rho_m}{\rho_f})U_m \tag{6}$$

$$\tau_f = (\frac{\rho_f}{\rho_m})(\frac{L_f}{L_m})^3(\frac{U_f}{U_m})^2\tau_m \tag{7}$$

where the subscript f refers to the full-size unit and m defines the modeled replica. τ is the torque output, ρ is the density of air, L is the length scale, U is the wind velocity and μ is the fluid dynamic viscosity.

Equation 6 and 7 can be written as the dimensionless components of velocity and torque, respectively, in Equations 8 and 9.

$$\bar{U} = \frac{UL\rho}{\mu} \tag{8}$$

$$\bar{\tau} = \frac{\tau}{\rho U^2 L^3} \tag{9}$$

The non-dimensional velocity is more commonly referred to as Reynold's number (Re). Given the imperfect scaling from the 2.5-inch to 5-inch replicas, due to the constraint of the output shaft diameter, the length scale was defined by the diameter of the turbine. These turbine diameters were Ø = 1.535-inches for the 2.5-inch tall turbine and Ø = 2.521-inches for the 5-inch tall turbine. Using these length scales for L, in Equations 8 and 9, along with $\rho = 1.2047 kg/m^3$ and $\mu =$

1.8205e-5$kg/(ms)$, for air at an ambient temperature of 20°C, we can plot the non-dimensional torque versus Reynold's number (Figure 9). Taking the average of the torque values above a Reynold's number of 20,000, yields a non-dimensional torque coefficient of 0.108.

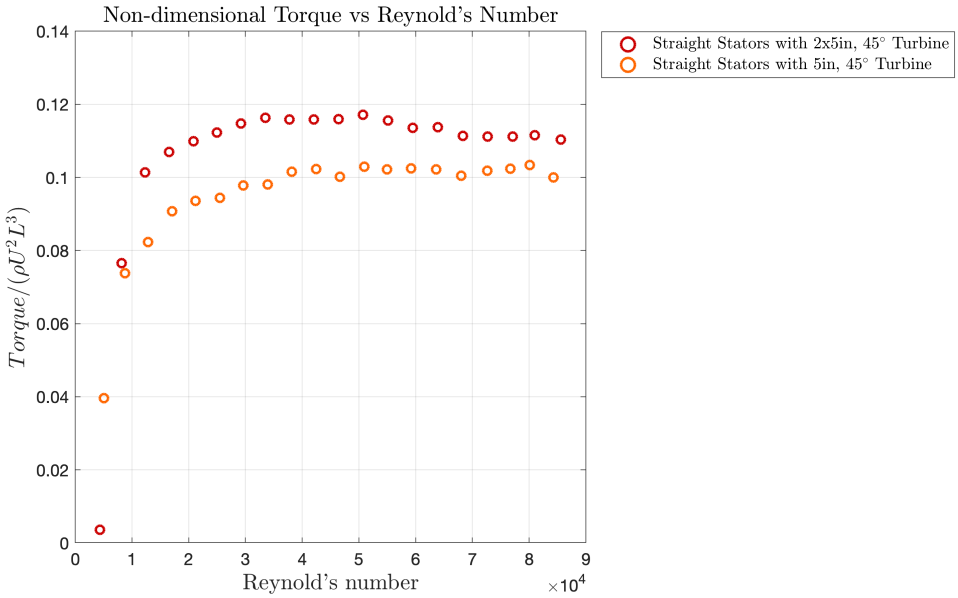


Figure 60: Non-dimensional output torque vs Reynold's number

This results in the final scaling equation shown in Equation 10. Applying this formula to the velocity data attained during testing of the 2.5 and 5-inch replicas shows good correlation to the corresponding torque data from the experiment (Figure 61).

$$\bar{\tau} = 0.108\rho U^2 L^3 \tag{10}$$

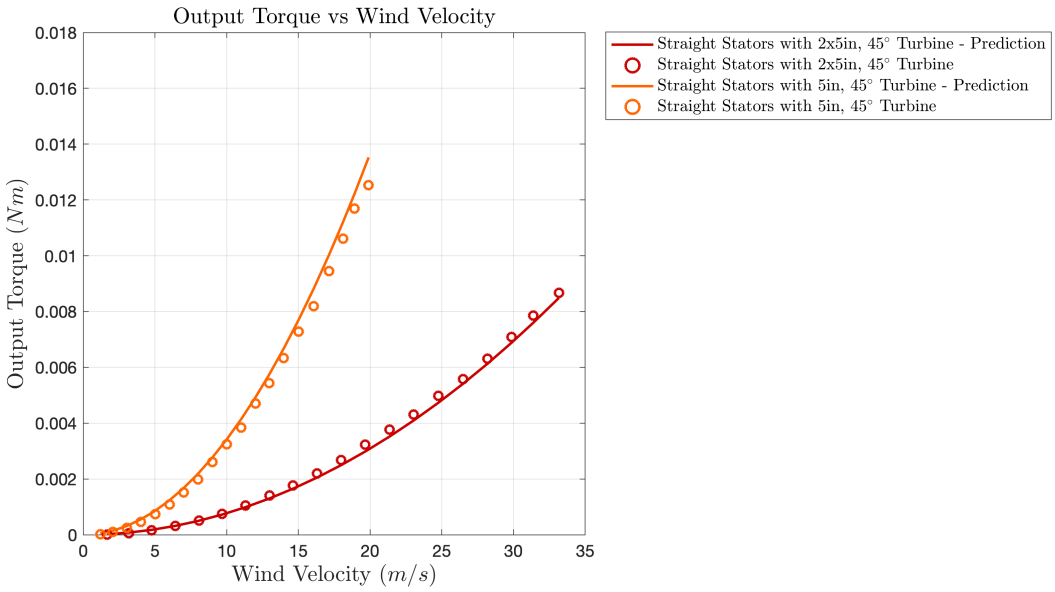


Figure 61: Output torque vs wind velocity for experiment and scaling equation prediction

With confidence established in the torque scaling equation from the ability to match the experimental data of the scale turbines, the equation was applied to the 7-foot tall prototype with 54-inch diameter turbine. The prediction can be seen plotted in (Figure 62).

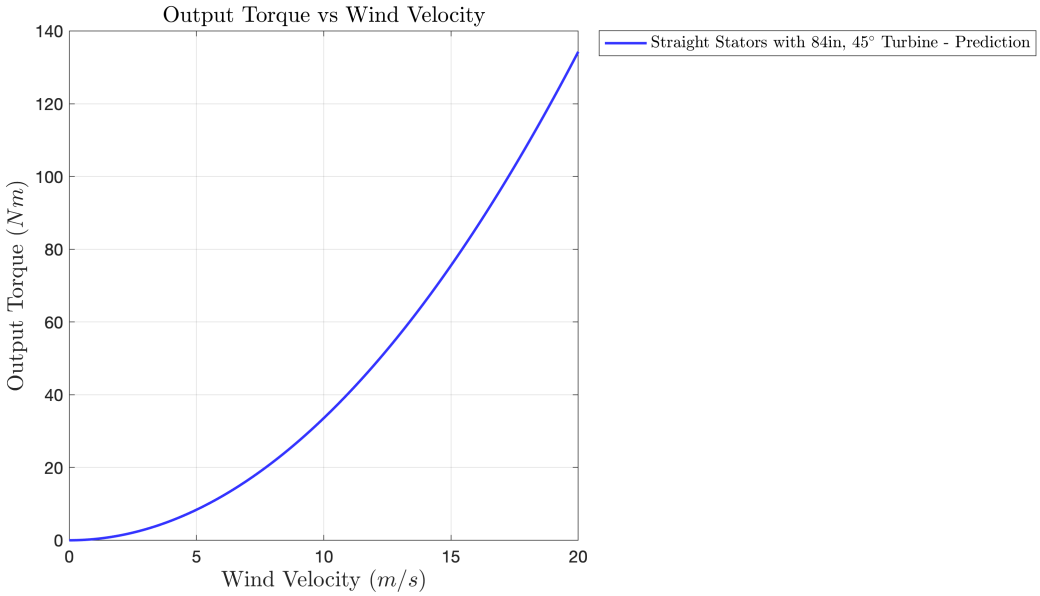


Figure 62: Output torque vs wind velocity for scaling equation prediction of 7-foot prototype

Finally, to compare the scaling equation prediction to the actual output torque of the 7-foot prototype, the experimental results are plotted with projected torque in Figure 63. Given that the lowest performing stator and turbine combinations were scaled from the replicas, the results from the 7-foot prototype predominantly outperforming the curve in all recorded data suggest the 7-foot prototype can accurately be described by the scaled replica data.

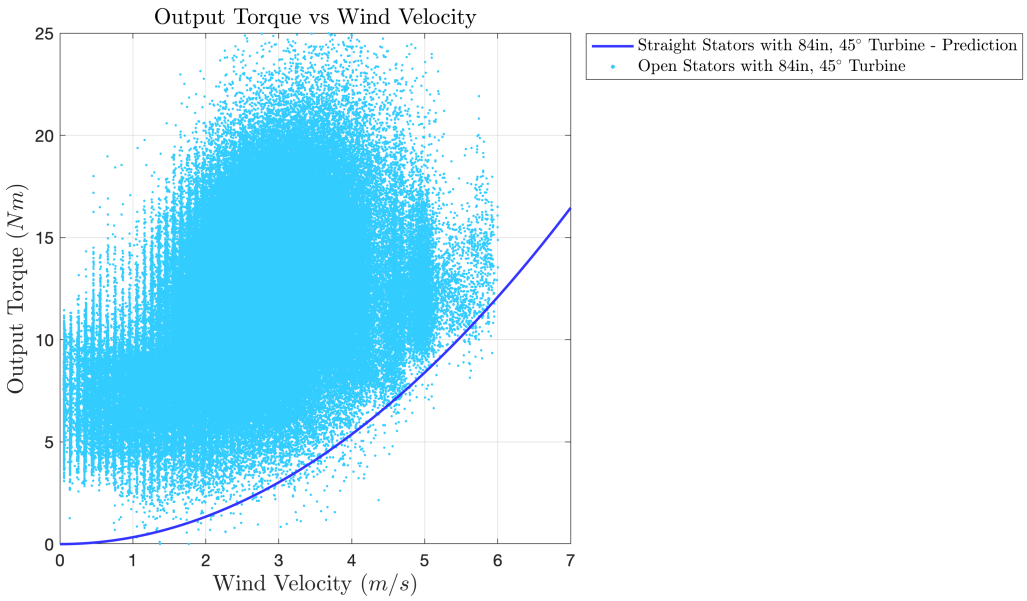


Figure 63: Output torque vs wind velocity for scaling equation prediction and experimental data for the 7-foot prototype

CHAPTER 9

Conclusion

9.1 Turbine Geometry

This research has demonstrated the utility of using scale replica wind tunnel testing to predict optimal turbine geometry. The chosen turbine features were varied and tested for performance in a number of conditions. These included, static torque output with variable wind velocity and turbine offset, angular displacement versus wind velocity and angular displacement of a turbine released from rest in a constant flow field. The transition from a straight stator to one of a 45° top-to-bottom helix showed dramatic improvement in both static and dynamic testing. However, the addition of an open stator in place of a straight stator showed only marginal benefits.

These results give rise to questions such as whether or not the benefits of opening stators is worth the cost of manufacture and risk of malfunction due to added complexity. While the performance results make this question easier to answer for the helical turbine, data is provided to make an informed decision in both circumstances.

9.2 Turbine Scaling

With the requirement of predicting a turbine height for producing an output torque capable of starting a preselected generator, a dimension was determined. The predicted height showed success in full prototype testing by successfully starting an unloaded 5 kW generator at an average of 1.99 m/s of wind velocity. Furthermore, the low values of the torque produced by the 7-foot turbine, closely tracked the prediction made by the scaling equation. These results built confidence in the predictability of turbine scaling by use of non-dimensionl analysis.

9.3 Future Work

While the fundamental questions of the HIPS turbine function have been answered, nothing has been done to predict the actual power output of the device. This gives way to further research of scaling power output using model replicas, along with actual measurements from the 7-foot prototype. This research would define the true usability of the design and provide a basis for comparison to existing production units. Power output and efficiency data will demonstrate the potential benefit associated with a fixed stator, vertical axis turbine design.

Additionally, the construction of the 7-foot turbine was for proof-of-concept and validation of scaling predictions, yet structural rigidity was not tested. The design may need to be modeled using finite element methods to determine the stability of the current design in extreme weather conditions. Following the analysis, the design may need to be refined to accommodate harsh weather. Material degradation should also be studied given the potential of corrosive sea air contacting the assembly.

LIST OF REFERENCES

Alejandro Franco, J., Carlos Jauregui, J., and Toledano-Ayala, M. (2015). Optimizing wind turbine efficiency by deformable structures in smart blades. *Journal of Energy Resources Technology*, 137(5).

Archibald, E. D. (1883). The increase in the velocity of the wind with the altitude. *Nature*, 27(689):243–245.

Burlando, M., Ricci, A., Freda, A., and Repetto, M. (2015). Numerical and experimental methods to investigate the behaviour of vertical-axis wind turbines with stators. *Journal of Wind Engineering & Industrial Aerodynamics*, 144:125–133.

De Lellis, M., Reginatto, R., Saraiva, R., and Trofino, A. (2018). The betz limit applied to airborne wind energy. *Renewable Energy*, 127:32–40.

Ebert, P. and Wood, D. (1997). Observations of the starting behaviour of a small horizontalaxis wind turbine. *Renewable Energy*, 12(3):245–257.

Farthing, S. (2013). Betz limit not an exact optimum. *Wind Engineering*, 37(1):105–109.

Foken, T. (2006). 50 years of the monin–obukhov similarity theory. *Boundary-Layer Meteorology*, 119(3):431–447.

Franković, B. and Vrsalović, I. (2001). New high profitable wind turbines. *Renewable Energy*, 24(3-4):491–499.

Giahi, M. H. and Jafarian Dehkordi, A. (2016). Investigating the influence of dimensional scaling on aerodynamic characteristics of wind turbine using cfd simulation. *Renewable Energy*, 97:162–168.

Jafarnejadsani, H., Pieper, J., and Ehlers, J. (2013). Adaptive control of a variable-speed variable-pitch wind turbine using radial-basis function neural network. *IEEE Transactions on Control Systems Technology*, 21(6):2264–2272.

Nelson, D. (2008). *The Penguin Dictionary Of Mathematics*. Penguin.

Rolin, V. F.-C. and Porté-Agel, F. (2018). Experimental investigation of vertical-axis wind-turbine wakes in boundary layer flow. *Renewable Energy*, 118:1–13.

Worasinchai, S., Ingram, G. L., and Dominy, R. G. (2012). Effects of wind turbine starting capability on energy yield. *Journal of Engineering for Gas Turbines and Power*, 134(4).

Wright, A. and Wood, D. (2004). The starting and low wind speed behaviour of a small horizontal axis wind turbine. *Journal of Wind Engineering & Industrial Aerodynamics*, 92(14-15):1265–1279.

Zamani, M., Maghrebi, M. J., and Varedi, S. R. (2016). Starting torque improvement using j-shaped straight-bladed darrieus vertical axis wind turbine by means of numerical simulation. *Renewable Energy*, 95:109–126.

APPENDIX A

Visual Basic GUI and DAQ Control

```
'Milo Ferrazzoli
'HIPS Turbine Data Collection
'July 8, 2016

Imports System.IO.Ports
Imports System.Windows.Forms.DataVisualization.Charting

Public Class turbine_data
    Dim comSet As String
    Dim dataIn As String
    Dim dataInArray() As String
    Dim dataOut As String = Nothing
    Dim comPort As SerialPort
    Dim fileLocation$ = "ss_45t_mini_2000"
    Dim getDataInterval# = 100 'microseconds
    Dim lastDataTime#
    Dim dataInc% = 0
    Dim forceChartMin% = 0
    Dim forceChartMax% = 75
    Dim pressChartMin% = 0
    Dim pressChartMax% = 500
    Dim baudRate& = 230400
    Dim dataVars% = 3 'Number of variables to write to file
    Dim scrollPoints% = 110
    Dim startProgram As Boolean = False
    Dim count As Int64
    Dim state$
    Dim NextState As String = "Start"
    Delegate Sub SetTextCallback(ByVal [text] As String)
    Delegate Sub SetDoubleTextCallback(ByVal [text] As String, ByVal [text] As String)
    Dim isRotating$
    Dim stageLocation#, currentLoc#
    Dim dataStorage(dataVars, dataInc) As Double
    Dim dataInTimer As New PerformanceTimer
    Dim dataOldTime&
    Dim dataWriteOldTime&
    Dim writeDataInterval& = 1000 'how often a data set is logged in microseconds.

    Dim leverArm# = 0.035 '35mm load cell bending length
    Dim accelGravity# = 9.80665

    Dim arduinoTime$ = 0
    Dim arduinoForce$ = 0
    Dim arduinoPressure$ = 0

    Dim runCmd$ = 0
    Dim tareCmd$ = 0
    Dim resetCmd$ = 0
```

```
49
50      Dim captureData As Boolean = False
51      Dim writeData As Boolean = False
52
53      Dim timePrint#
54      Dim forcePrint#
55      Dim pressPrint#
56
57      Dim cellScaleFactor# = 187.083 '131
58      Dim cellTareValue# = 59.2 '87
59      Dim pressScaleFactor# = 23.12
60      Dim pressTareValue# = 85.8 '86.82
61      Dim kinViscosityAir = 1.225
62
63
64      Private Sub ferrazzoli_pa6_Load(sender As Object, e As EventArgs) Handles MyBase.Load
65          initializeAll()
66      End Sub
67
68      Private Sub getFileName()
69          fileLocation = "C:\Users\Public\Documents\"" + tbFileName.Text + ".txt"
70      End Sub
71      Private Sub initializeAll()
72          comSet = Nothing
73
74          tBDataInt.Text = writeDataInterval
75          lbCapture.Visible = False
76          tbFileName.Text = fileLocation
77
78          getSerialPorts()
79          initilizeChart()
80
81          dataInTimer.StartTimer()
82          serialWorker.RunWorkerAsync()
83      End Sub
84
85      Private Sub serialWorker_DoWork(sender As Object, e As System.ComponentModel.true
        DoWorkEventArgs) Handles serialWorker.DoWork
86          If serialWorker.CancellationPending Then
87              e.Cancel = True
88          Else
89              Do While (2 > 1)
90                  count += 1
91                  controlTask()
92              Loop
93          End If
94      End Sub
95
96      Private Sub controlTask()
97          state = NextState
98          Select Case state
99
100             Case "Start"
101                 If (startProgram = True) Then
```

```
                    If (comSet IsNot Nothing) Then
                        NextState = "Wait"
                    Else
                        MsgBox("Please choose a COM port")
                        startProgram = False
                    End If
                End If

            Case "Wait"
                checkArduinoData()

            Case "SendData"
                sendSerial(dataOut)
                dataOut = Nothing
                NextState = "Wait"

            Case "ReadData"
                dataIn = getSerial()
                parseDataIn(dataIn)
                NextState = "UpdateChart"

            Case "UpdateChart"
                upDateChart()
                NextState = "Wait"
        End Select
    End Sub

    Private Sub upDateChart()
        Try
            timePrint = Math.Round((Convert.ToDouble(arduinoTime) / 1000), 3)
            forcePrint = Math.Round(((Convert.ToDouble(arduinoForce) / cellScaleFactor) - true
cellTareValue), 2)
            'forcePrint = Math.Round(((Convert.ToDouble(arduinoTorque) * accelGravity) * true
leverArm), 3)
            'pressPrint = Math.Round((Math.Sqrt((2 * ((Convert.ToDouble(arduinoPressure) / true
pressScaleFactor) - pressTareValue)) / kinViscosityAir)), 3)
            pressPrint = Math.Round(((Convert.ToDouble(arduinoPressure) / pressScaleFactor) true
- pressTareValue), 2)
        Catch
        End Try

        updateForceChart(timePrint, forcePrint)
        updatePressChart(timePrint, pressPrint)
        displayArduinoTime(timePrint)
        displayArduinoTorque(forcePrint)
        displayArduinoPressure(pressPrint)
        updateForceChartSize(timePrint)
        updatePressChartSize(timePrint)

        manageFileData()
    End Sub

    Private Sub manageFileData()
        If (lastDataTime <> timePrint) Then
```

```
            If captureData = True Then
                If (checkTime() - dataWriteOldTime) > writeDataInterval Then
                    dataInTimer.ReadCurrentTimer()
                    dataWriteOldTime = checkTime()
                    Try
                        ReDim Preserve dataStorage(dataVars, dataInc + 1)
                        dataStorage(0, dataInc) = timePrint
                        dataStorage(1, dataInc) = forcePrint
                        dataStorage(2, dataInc) = pressPrint
                        dataInc += 1
                    Catch
                    End Try
                End If
                If writeData = True Then
                    writeDatatoFile()
                    Array.Clear(dataStorage, 0, dataStorage.Length)
                    writeData = False
                    captureData = False
                End If
            End If
        End If

        lastDataTime = timePrint
    End Sub

    Private Sub getSerialPorts()
        For Each comPortName As String In My.Computer.Ports.SerialPortNames
            cBComList.Items.Add(comPortName)
        Next
    End Sub

    Private Sub cBComList_SelectedIndexChanged(sender As Object, e As EventArgs) Handles true
     cBComList.SelectedIndexChanged
        comSet = cBComList.SelectedItem
        comPort = New SerialPort(comSet)
        comPort.DataBits = 8
        comPort.BaudRate = baudRate
        comPort.Parity = Parity.Even
        comPort.StopBits = StopBits.One
        comPort.Open()
    End Sub

    Private Sub initilizeChart()
        chartForceData.ChartAreas(0).AxisY.Minimum = forceChartMin
        chartForceData.ChartAreas(0).AxisY.Maximum = forceChartMax
        chartForceData.Series(0).ChartType = SeriesChartType.FastLine
        chartForceData.ChartAreas(0).AxisX.Title = "Time (s)"
        chartForceData.ChartAreas(0).AxisY.Title = "Load Cell Force (g)"

        chartPressData.ChartAreas(0).AxisY.Minimum = pressChartMin
        chartPressData.ChartAreas(0).AxisY.Maximum = pressChartMax
        chartPressData.Series(0).ChartType = SeriesChartType.FastLine
        chartPressData.ChartAreas(0).AxisX.Title = "Time (s)"
        chartPressData.ChartAreas(0).AxisY.Title = "Pitot Tube Pressure (Pa)"
```

```
205     End Sub
206
207     Private Sub checkArduinoData()
208         If dataOut IsNot Nothing Then
209             NextState = "SendData"
210         ElseIf (checkTime() - dataOldTime) > getDataInterval Then
211             NextState = "ReadData"
212             dataInTimer.ReadCurrentTimer()
213             dataOldTime = checkTime()
214         End If
215         writeDataInterval = tBDataInt.Text
216     End Sub
217
218     Function checkTime() As Long
219         dataInTimer.ReadCurrentTimer()
220         checkTime = dataInTimer.TimeElapsed(PerformanceTimer.PerformanceValue.pvMicroSecond)
221     End Function
222
223     Sub sendSerial(ByVal data As String)
224         comPort.WriteLine(data)
225     End Sub
226
227     Function getSerial() As String
228         Dim incoming As String = Nothing
229         incoming = comPort.ReadLine()
230         Return incoming
231     End Function
232
233     Private Sub parseDataIn(ByVal data As String)
234         If data IsNot Nothing Then
235             Try
236                 dataInArray = data.Split(CType(" ", Char()))
237                 arduinoTime = dataInArray(0)
238                 arduinoForce = dataInArray(1)
239                 arduinoPressure = dataInArray(2)
240             Catch
241             End Try
242         End If
243     End Sub
244
245     Public Sub writeDatatoFile()
246         Dim file As System.IO.StreamWriter
247         file = My.Computer.FileSystem.OpenTextFileWriter(fileLocation, False)
248         For i% = 0 To dataInc
249             file.WriteLine(Str$(dataStorage(0, i)) + "," + Str$(dataStorage(1, i)) + "," + true
    Str$(dataStorage(2, i)))
250         Next
251         file.Close()
252     End Sub
253
254     Private Sub btnStart_Click(sender As Object, e As EventArgs) Handles btnStart.Click
255         startProgram = True
256         runCmd = 1
257         resetCmd = 1
```

```
258         dataOut = runCmd + " " + resetCmd
259         resetCmd = 0
260     End Sub
261 
262     Private Sub btnTare_Click(sender As Object, e As EventArgs) Handles btnTare.Click
263         cellTareValue = (Convert.ToDouble(arduinoForce) / cellScaleFactor)
264     End Sub
265 
266     Private Sub btnStop_Click(sender As Object, e As EventArgs) Handles btnStop.Click
267         runCmd = 0
268         dataOut = runCmd + " " + resetCmd
269     End Sub
270 
271     Private Sub btnTimeReset_Click(sender As Object, e As EventArgs) Handles btnTimeReset.true
        Click
272         resetCmd = 1
273         dataOut = runCmd + " " + resetCmd
274         resetCmd = 0
275     End Sub
276 
277     Private Sub btnCapture_Click(sender As Object, e As EventArgs) Handles btnCapture.Click
278         getFileName()
279         dataInc = 0
280         captureData = True
281         lbCapture.Visible = True
282     End Sub
283     Private Sub btnCapZero_Click(sender As Object, e As EventArgs) Handles btnCapZero.Click
284         getFileName()
285         dataInc = 0
286         captureData = True
287         resetCmd = 1
288         dataOut = runCmd + " " + resetCmd
289         resetCmd = 0
290         lbCapture.Visible = True
291     End Sub
292 
293     Private Sub btnEndCapture_Click(sender As Object, e As EventArgs) Handles btnEndCapture.true
        Click
294         writeData = True
295         lbCapture.Visible = False
296     End Sub
297 
298     Private Sub displayArduinoTime(ByVal [text] As String)
299         If Me.tBArdTime.InvokeRequired Then
300             Dim d As New SetTextCallback(AddressOf displayArduinoTime)
301             Me.Invoke(d, New Object() {[text]})
302         Else
303             Me.tBArdTime.Text = [text]
304         End If
305     End Sub
306 
307     Private Sub displayArduinoTorque(ByVal [text] As String)
308         If Me.tBArdTorq.InvokeRequired Then
309             Dim d As New SetTextCallback(AddressOf displayArduinoTorque)
```

```
310             Me.Invoke(d, New Object() {[text]})
311         Else
312             Me.tBArdTorq.Text = [text]
313         End If
314     End Sub
315 
316     Private Sub displayArduinoPressure(ByVal [text] As String)
317         If Me.tBArdPress.InvokeRequired Then
318             Dim d As New SetTextCallback(AddressOf displayArduinoPressure)
319             Me.Invoke(d, New Object() {[text]})
320         Else
321             Me.tBArdPress.Text = [text]
322         End If
323     End Sub
324 
325     Private Sub updateForceChart(ByVal [text] As String, ByVal [text2] As String)
326         If Me.chartForceData.InvokeRequired Then
327             Dim d As New SetDoubleTextCallback(AddressOf updateForceChart)
328             Me.Invoke(d, New Object() {[text], [text2]})
329         Else
330             Me.chartForceData.Series(0).Points.AddXY([text], [text2])
331         End If
332 
333     End Sub
334 
335     Private Sub updatePressChart(ByVal [text] As String, ByVal [text2] As String)
336         If Me.chartPressData.InvokeRequired Then
337             Dim d As New SetDoubleTextCallback(AddressOf updatePressChart)
338             Me.Invoke(d, New Object() {[text], [text2]})
339         Else
340             Me.chartPressData.Series(0).Points.AddXY([text], [text2])
341         End If
342 
343     End Sub
344 
345 
346     Private Sub updateForceChartSize(ByVal [text] As String)
347         If Me.chartForceData.InvokeRequired Then
348             Dim d As New SetTextCallback(AddressOf updateForceChartSize)
349             Me.Invoke(d, New Object() {[text]})
350         Else
351             If ((chartForceData.ChartAreas(0).AxisX.Maximum - chartForceData.ChartAreas(0).true
    AxisX.Minimum) > scrollPoints) Then
352                 chartForceData.ChartAreas(0).AxisX.Minimum += 1
353             End If
354         End If
355     End Sub
356 
357     Private Sub updatePressChartSize(ByVal [text] As String)
358         If Me.chartPressData.InvokeRequired Then
359             Dim d As New SetTextCallback(AddressOf updatePressChartSize)
360             Me.Invoke(d, New Object() {[text]})
361         Else
362             If ((chartPressData.ChartAreas(0).AxisX.Maximum - chartPressData.ChartAreas(0).true
```

```
    AxisX.Minimum) > scrollPoints) Then
363                 chartPressData.ChartAreas(0).AxisX.Minimum += 1
364             End If
365         End If
366     End Sub
367 End Class
```

APPENDIX B

Arduino C++ Hardware Control

```
#include <HX711.h>
#include <Arduino.h>
#include <stdlib.h>


const long baudRate = 230400;
const char endLine = 10;

int runCmd = 0;
int tareCmd = 0;
int timeInc = 32;
//int timeInc = 10;
int timeLoop = 0;
long timeMS = 0;

int state;
const int runReady = 1;
const int tare = 2;
const int checkTime = 3;
const int getData = 4;
const int serialSend = 5;

int nextState = runReady;

//int calFac = -5090;
int calFac = -1270;
int cellCLK = 2;
int cellDT = 3;
int pressData = A0;
HX711 cell(cellDT, cellCLK);

double cellData;
int pressureData;

int timeReset = 1;
boolean runReset = true;

ISR(TIMER2_OVF_vect) {
  controlRoutine();
}

void setup()
{
  TCCR2A = 0x00;
  TCNT2 = 0x06;
  TCCR2B = 0x04;
  TIMSK2 = 0x01;
  Serial.begin(baudRate, SERIAL_8E1);
```

```
  loadCellSetup();
}

void loop()
{
getSerial();
}
////////////////////////////////////////////////////////////////////////////////CONTROL ROUTINE


void controlRoutine() {

  state = nextState;
  switch (state) {
    //-------------------------------------------------------------------------------ready State
    case runReady:
      //getSerial();
      if (tareCmd == 1)
        nextState = tare;
      else if (runCmd == 1) {
        nextState = checkTime;
        if (timeReset == 1 || runReset == true) {
          timeMS = -3;
          timeReset = 0;
          runReset = false;
        }
      }
      else if (runCmd == 0) {
        timeReset = 1;
        runReset = true;
      }
      break;
    //-------------------------------------------------------------------------------tare State
    case tare:
      loadCellSetup();
      nextState = runReady;
      tareCmd = 0;
      break;
    //-------------------------------------------------------------------------checkTime State
    case checkTime:
      if (timeLoop >= timeInc) {
        nextState = getData;
        timeLoop = 0;
      }
      else
        nextState = runReady;
      break;
    //-------------------------------------------------------------------------getCell State
    case getData:
      getCellData();
      getPressureData();
      nextState = serialSend;
      break;
    //-------------------------------------------------------------------------sendData State
```

```
    case serialSend:
      sendData();
      nextState = runReady;
      break;
  }
  timeMS += 1;
  timeLoop += 1;
  //if (timeMS % 10 == 0)
  //Serial.println(timeMS);
}

void sendData() {
  Serial.print(timeMS);
  Serial.print(" ");
  Serial.print(cellData);
  Serial.print(" ");
  Serial.print(pressureData);
  Serial.print(" ");
  Serial.print(endLine);
}

void getCellData() {
  cellData = cell.get_units(), 2;
}

void getPressureData() {
  pressureData = analogRead(pressData);
}

void loadCellSetup() {
  cell.set_gain(32);
  cell.set_scale(calFac);
  cell.tare();
}

void getSerial() {
  if (Serial.available() > 0) {
    String cmdString = Serial.readStringUntil(endLine);
    char cmdArray[15];
    cmdString.toCharArray(cmdArray, 15);

    runCmd = atof(strtok(cmdArray, " "));
    timeReset = atof(strtok(0, " "));
    tareCmd = atof(strtok(0, " "));
  }
}
```

APPENDIX C

Raw Data Plots

C.1 Torque Versus Wind Velocity of 5-inch Replica

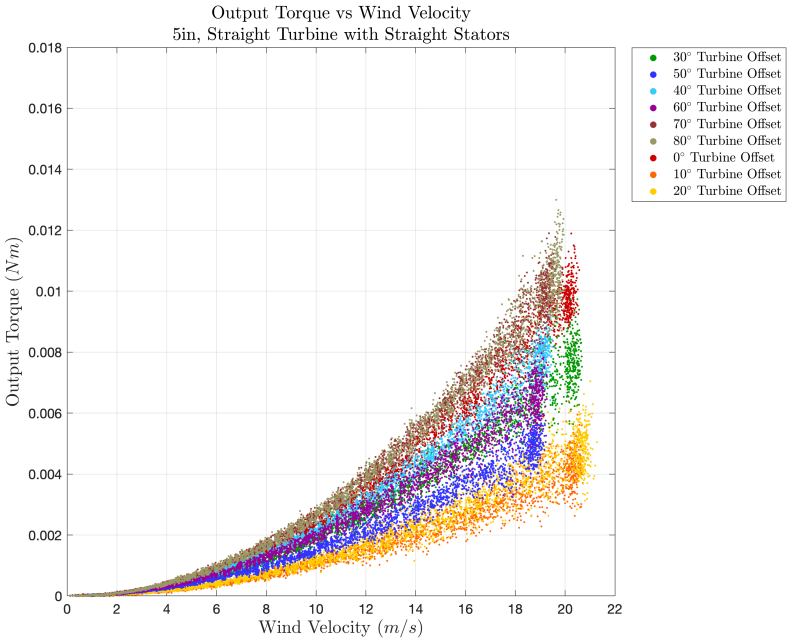


Figure C.64: Output torque vs wind velocity for 5-inch, straight turbine with straight stators

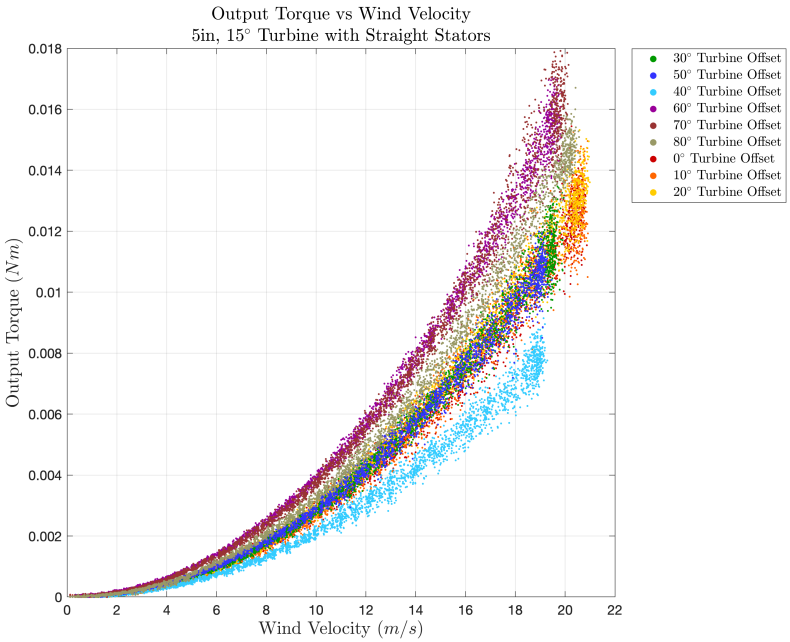


Figure C.65: Output torque vs wind velocity for 5-inch, 15° turbine with straight stators

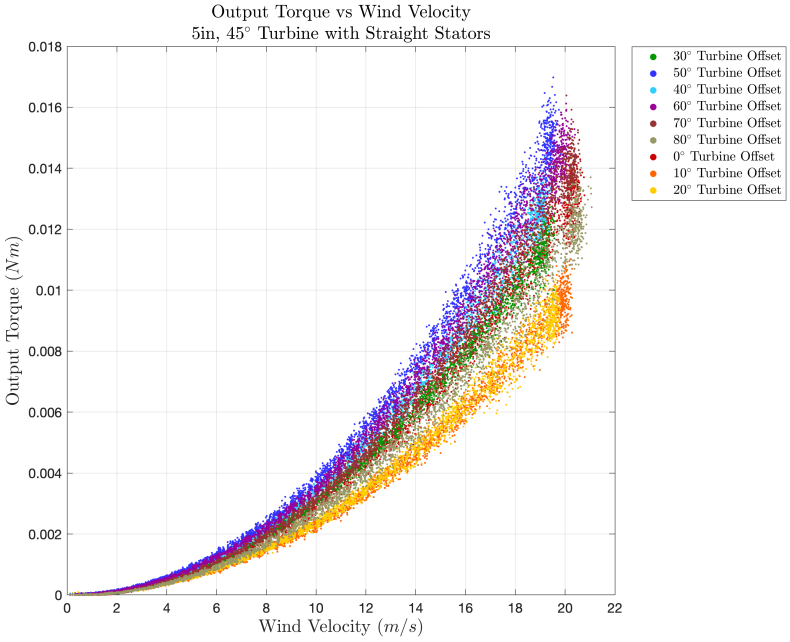


Figure C.66: Output torque vs wind velocity for 5-inch, 45° turbine with straight stators

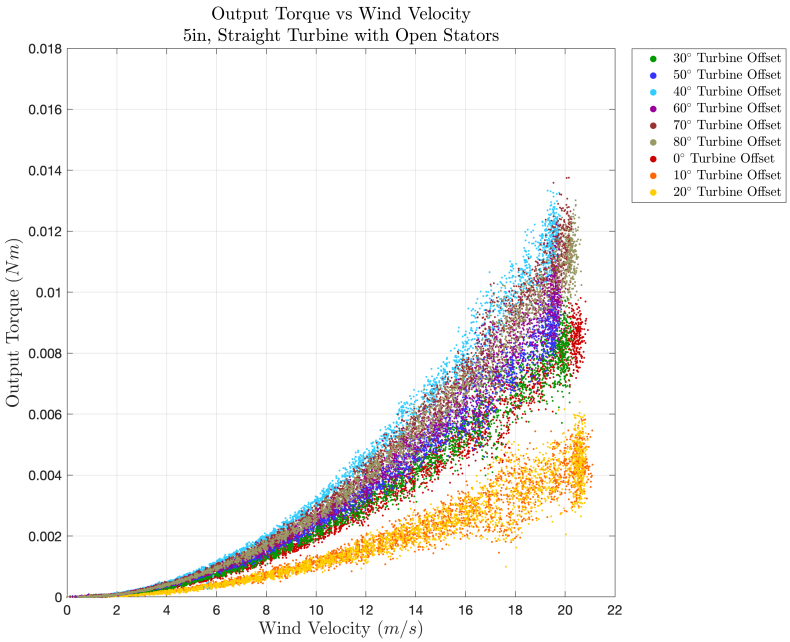


Figure C.67: Output torque vs wind velocity for 5-inch, straight turbine with open stators

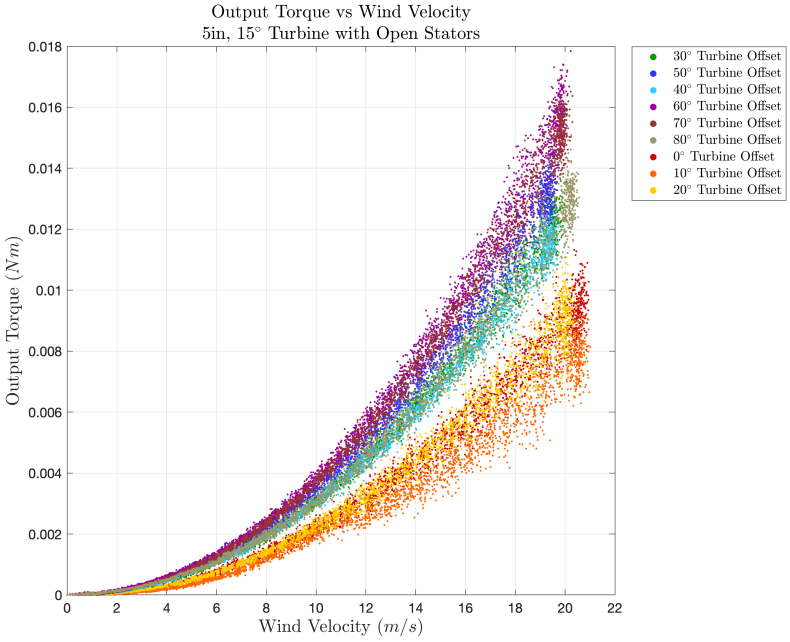


Figure C.68: Output torque vs wind velocity for 5-inch, 15° turbine with open stators

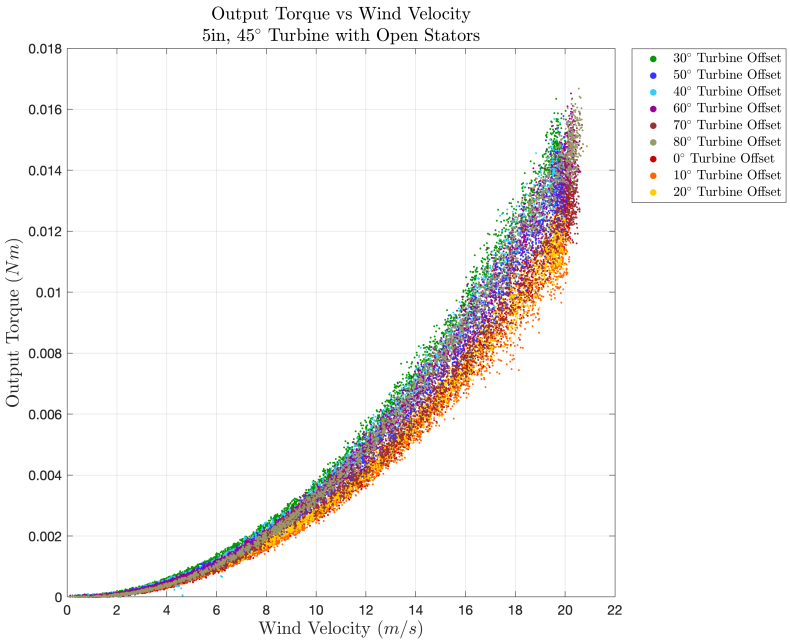


Figure C.69: Output torque vs wind velocity for 5-inch, 45° turbine with open stators

C.2 Torque Versus Wind Velocity of 5-inch Replica with Offset Data Averaged

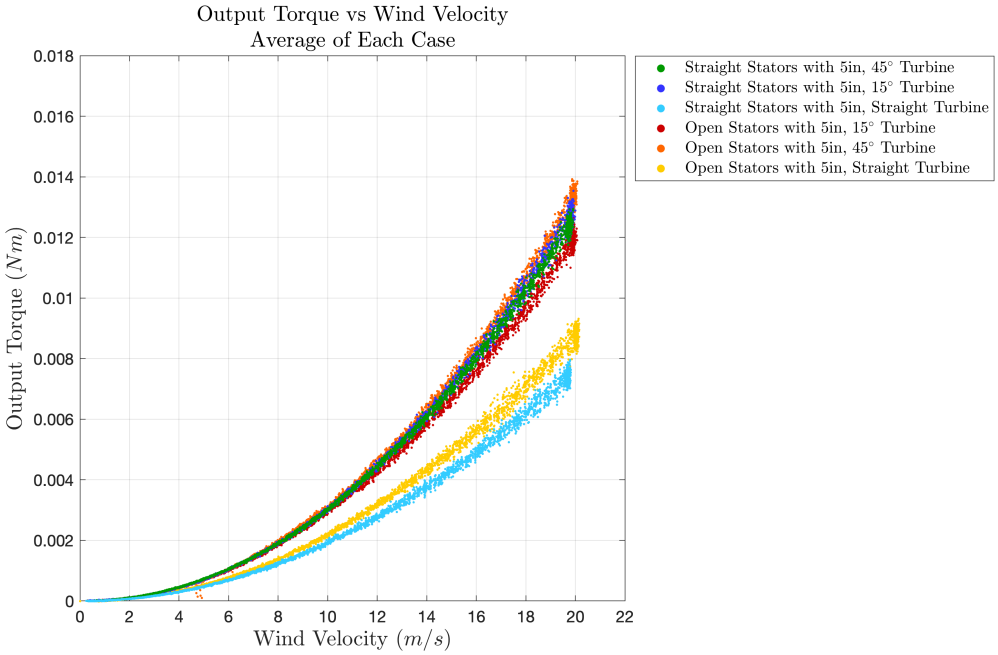


Figure C.70: Output torque vs wind velocity of values averaged for all offsets of each turbine and stator configuration

C.3 Torque Versus Wind Velocity of 5-inch Replica with Worst Performing Offset

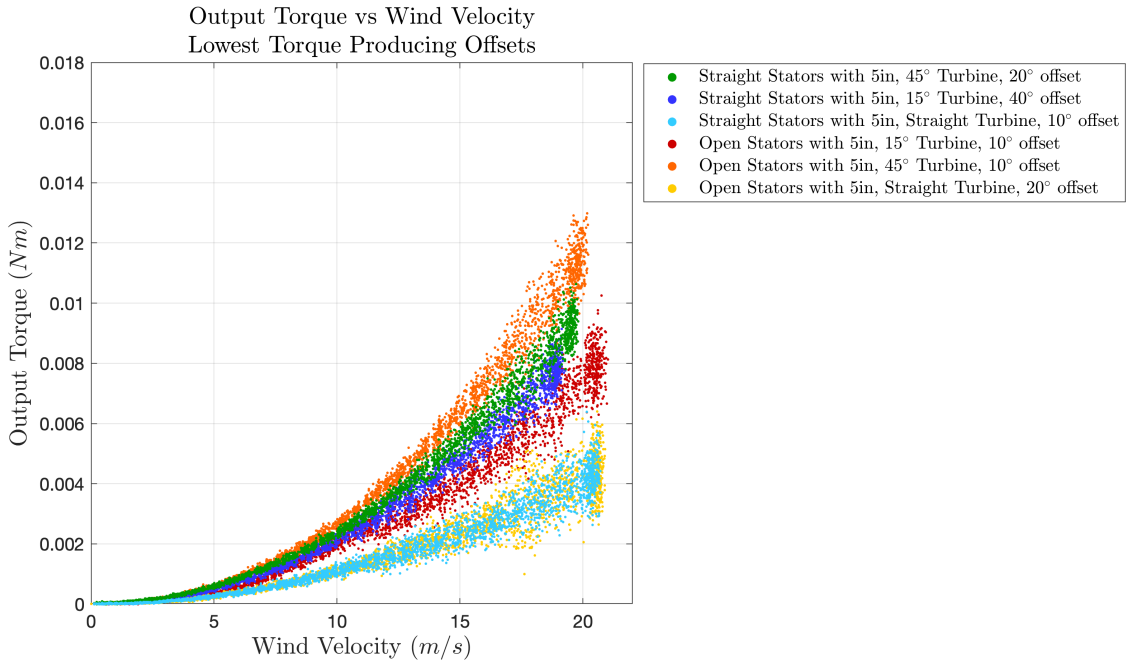


Figure C.71: Output torque vs wind velocity of the worst performing offset angles for each turbine and stator configuration

BIBLIOGRAPHY

Alejandro Franco, J., Carlos Jauregui, J., and Toledano-Ayala, M., "Optimizing wind turbine efficiency by deformable structures in smart blades," *Journal of Energy Resources Technology*, vol. 137, no. 5, 2015.

Archibald, E. D., "The increase in the velocity of the wind with the altitude," *Nature*, vol. 27, no. 689, pp. 243–245, 1883.

Burlando, M., Ricci, A., Freda, A., and Repetto, M., "Numerical and experimental methods to investigate the behaviour of vertical-axis wind turbines with stators," *Journal of Wind Engineering & Industrial Aerodynamics*, vol. 144, pp. 125–133, 2015.

De Lellis, M., Reginatto, R., Saraiva, R., and Trofino, A., "The betz limit applied to airborne wind energy," *Renewable Energy*, vol. 127, pp. 32–40, 2018.

Ebert, P. and Wood, D., "Observations of the starting behaviour of a small horizontalaxis wind turbine," *Renewable Energy*, vol. 12, no. 3, pp. 245–257, 1997.

Farthing, S., "Betz limit not an exact optimum," *Wind Engineering*, vol. 37, no. 1, pp. 105–109, 2013.

Foken, T., "50 years of the monin–obukhov similarity theory," *Boundary-Layer Meteorology*, vol. 119, no. 3, pp. 431–447, 2006.

Franković, B. and Vrsalović, I., "New high profitable wind turbines," *Renewable Energy*, vol. 24, no. 3-4, pp. 491–499, 2001.

Giahi, M. H. and Jafarian Dehkordi, A., "Investigating the influence of dimensional scaling on aerodynamic characteristics of wind turbine using cfd simulation," *Renewable Energy*, vol. 97, pp. 162–168, 2016.

Jafarnejadsani, H., Pieper, J., and Ehlers, J., "Adaptive control of a variable-speed variable-pitch wind turbine using radial-basis function neural network," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2264–2272, 2013.

Nelson, D., *The Penguin Dictionary Of Mathematics*. Penguin, 2008. [Online]. Available: http://search.credoreference.com/content/entry/penguinmath/the_penguin_dictionary_of_mathematics/0

Rolin, V. F.-C. and Porté-Agel, F., "Experimental investigation of vertical-axis wind-turbine wakes in boundary layer flow," *Renewable Energy*, vol. 118, pp. 1–13, 2018.

Worasinchai, S., Ingram, G. L., and Dominy, R. G., "Effects of wind turbine starting capability on energy yield," *Journal of Engineering for Gas Turbines and Power*, vol. 134, no. 4, 2012.

Wright, A. and Wood, D., "The starting and low wind speed behaviour of a small horizontal axis wind turbine," *Journal of Wind Engineering & Industrial Aerodynamics*, vol. 92, no. 14-15, pp. 1265–1279, 2004.

Zamani, M., Maghrebi, M. J., and Varedi, S. R., "Starting torque improvement using j-shaped straight-bladed darrieus vertical axis wind turbine by means of numerical simulation," *Renewable Energy*, vol. 95, pp. 109–126, 2016.